Garage Management System

COLLEGE NAME: NAVARASAM ARTS AND SCIENCE COLLEGE FOR WOMEN

COLLEGE CODE: asbru3a
TEAM ID: NM2025TMID27240
TEAM LEADER NAME: KARTHIYAYINI.TS

EMAIL: karthiyayinisivakumar2006@gmail.com

TEAM MEMBER1: SANTHIYA.R

EMAIL: rangarajansanthiya900@gmail.com

TEAM MEMBER2: GAYATHRI.M

EMAIL: gayu31284@gmail.com

TEAM MEMBER3: SOWMIYA.C

EMAIL: sowmiyachenniyappan@gmail.com

GARAGE MANAGEMENT SYSTEM

PROJECT OVERVIEW:

The Garage Management System project uses Salesforce CRM to build an application that manages the operations of vehicle service garages. The system connects customers, mechanics, inventory managers, and administrators to streamline service workflows. With the help of custom objects, automation, flows, and dashboards, the application handles customer details, vehicle records, service scheduling, spare parts tracking, and billing in a centralized platform. The process becomes efficient, transparent, and paperless, while customers receive timely updates and better service. This project demonstrates how technology can be applied to optimize garage management and improve customer satisfaction.

OBJECTIVES:

- To maintain accurate records of customers and their vehicles.
- To automate service scheduling, reminders, and job assignments.
- To track spare parts inventory and procurement.
- To generate invoices, estimates, and payment records efficiently.
- To improve customer engagement using Salesforce features (emails, SMS, service updates).
- To provide dashboards and reports for performance analysis and decision-making.

STUDENTS OUTCOMES

● **Hands-on Experience with Garage Management Automation:** Students gain practical skills in configuring Salesforce objects, automating service workflows, and managing real-time vehicle and inventory tracking.

● **Understanding of Project Lifecycle in Salesforce CRM:** Students learn the complete end-to-end process from requirement gathering to deployment, enhancing their ability to execute real-world Salesforce projects.

● **Enhanced Analytical and Problem-Solving Skills:** Students develop the ability to identify operational challenges in garage management, design solutions, and troubleshoot issues effectively.

● **Improved Collaboration Skills:** Students gain experience in teamwork, coordinating tasks such as requirement gathering, development, and testing.

● **Industry-Relevant Exposure:** Students get exposure to real-world use cases of Salesforce CRM in service management projects, preparing them for future career opportunities.

SYSTEM REQUIREMENTS

Hardware Requirements:

- Computer with minimum 4 GB RAM, Dual-core processor
- Stable internet connection

Software Requirements:

- Salesforce Developer Edition Org
- Modern Web Browser (e.g., Google Chrome, Firefox, Edge)

PURPOSE: To streamline and automate garage operations using Salesforce CRM for efficient service management and improved customer satisfaction.

PHASES 1

REQUIREMENTS ANALYSIS & PLANNING: -

The project requires a cloud-based Garage Management System to handle customer details, vehicle records, service scheduling, inventory, and billing. Functional and non-functional needs were identified, and modules were planned in Salesforce with clear milestones for design, development, testing, and deployment.

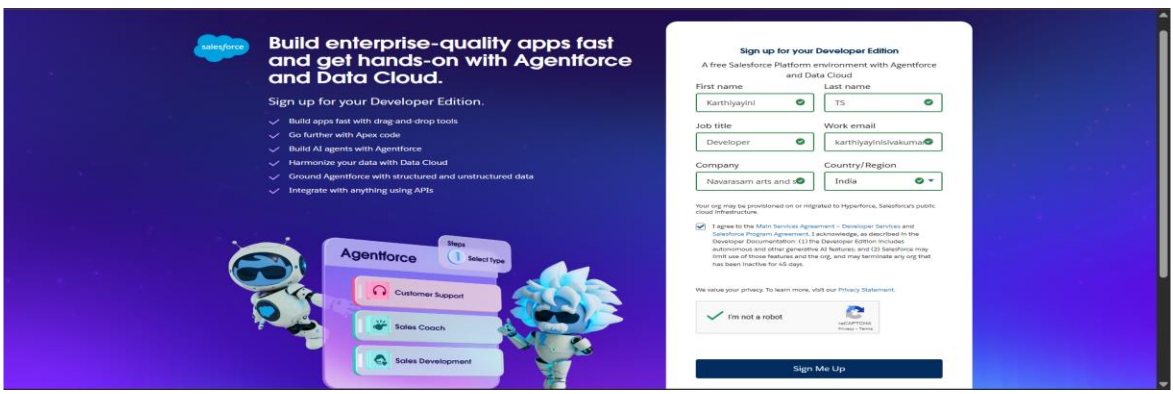
PHASE 2

DEVELOPMENT PHASE

MILESTONE 1: Salesforce developer account creation

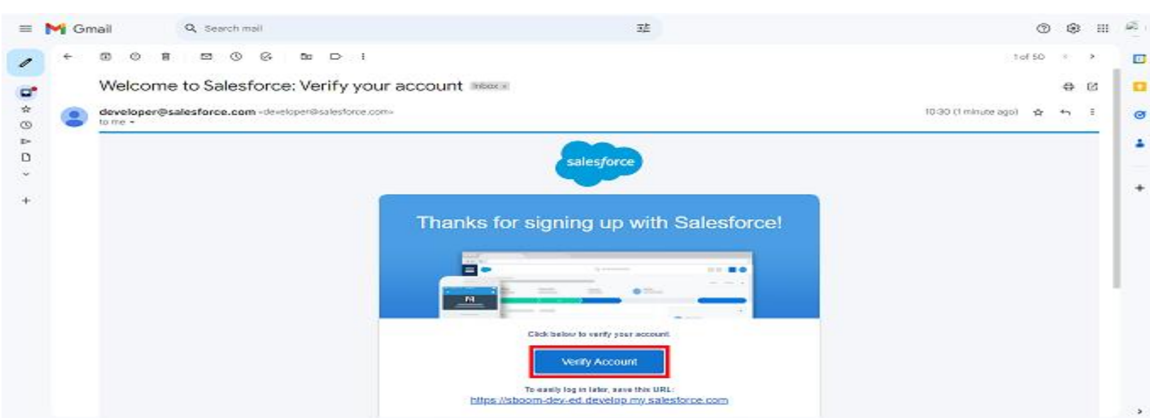
ACTIVITY 1: Creating Developer Account:

By using this URL: https://www.salesforce.com/form/developer-signup/?d=pb

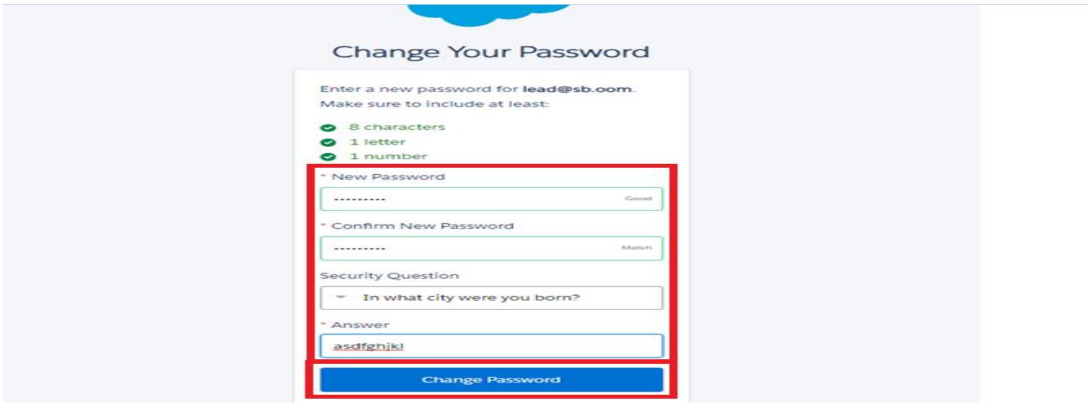


ACTIVITY 2: Account Activation

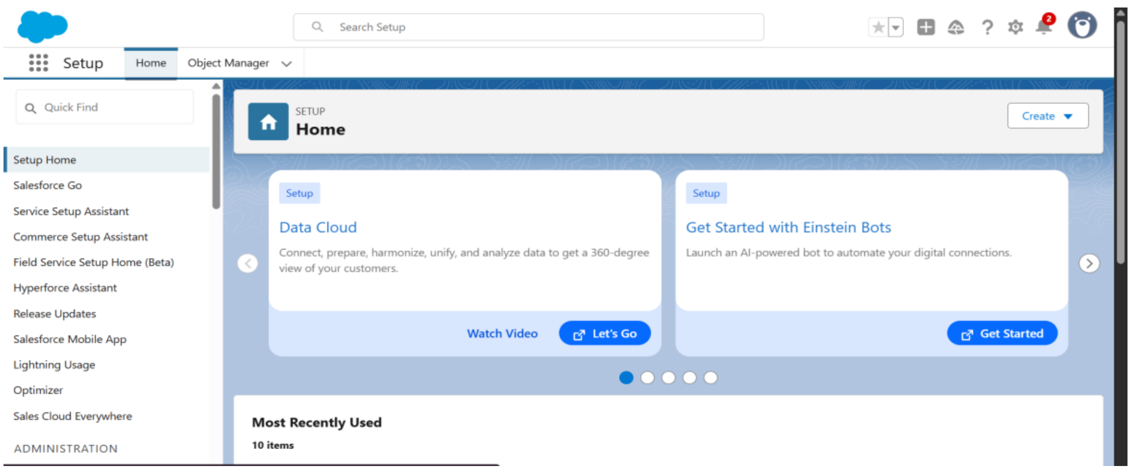
Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



Give a password and answer a security question and click on change password.

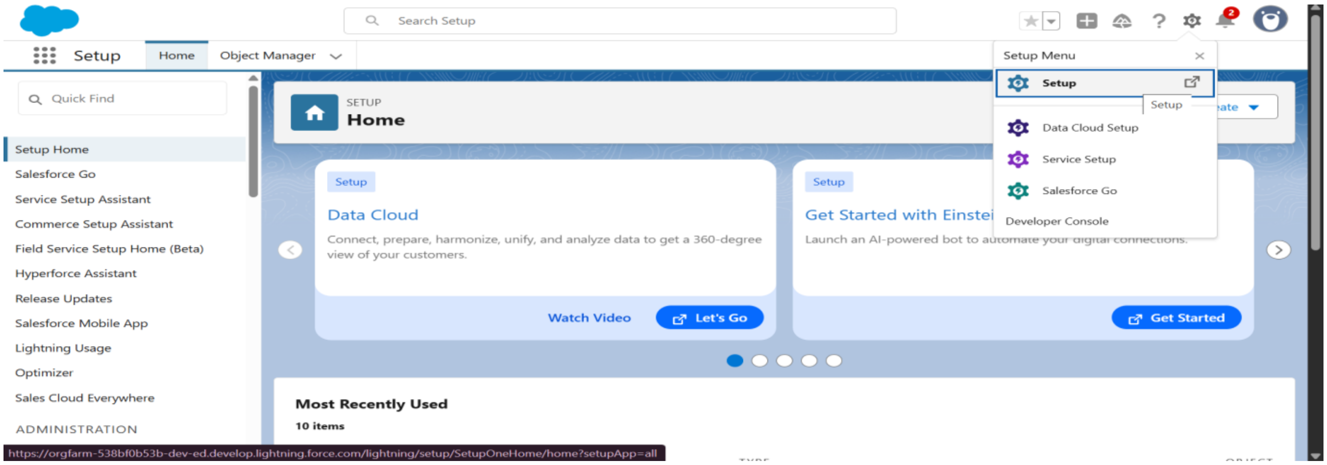


Then you will redirect to your salesforce setup page.



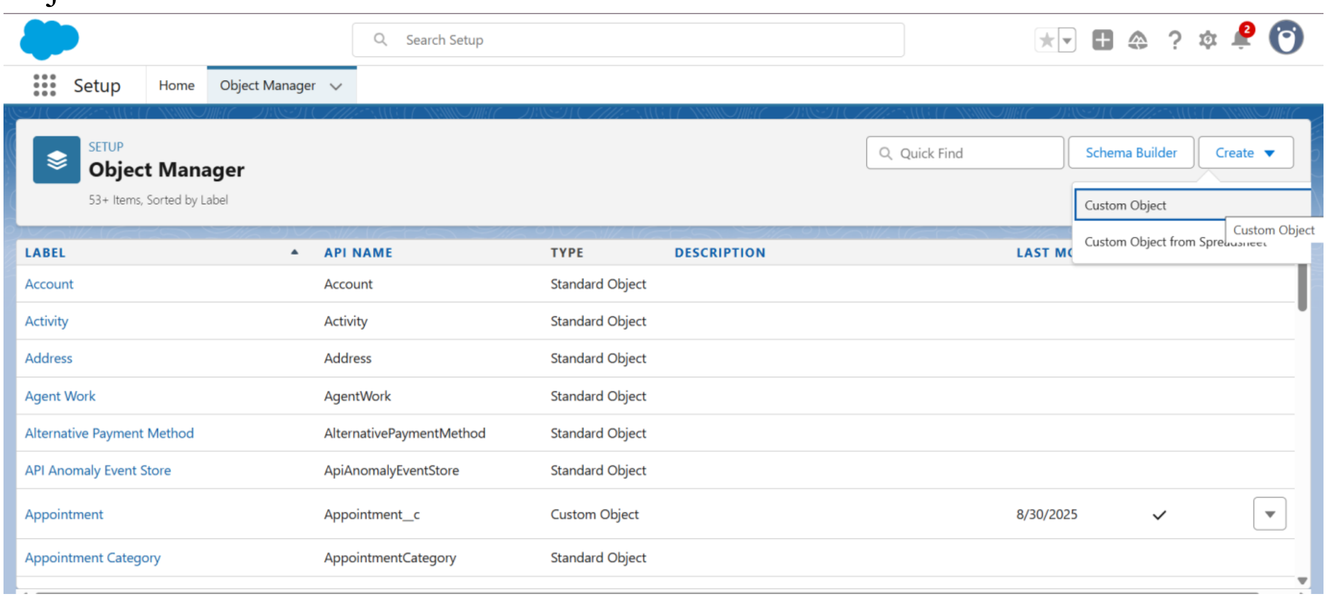
MILESTONE 2: Object

To Navigate to Setup page:
Click on gear icon? click setup.



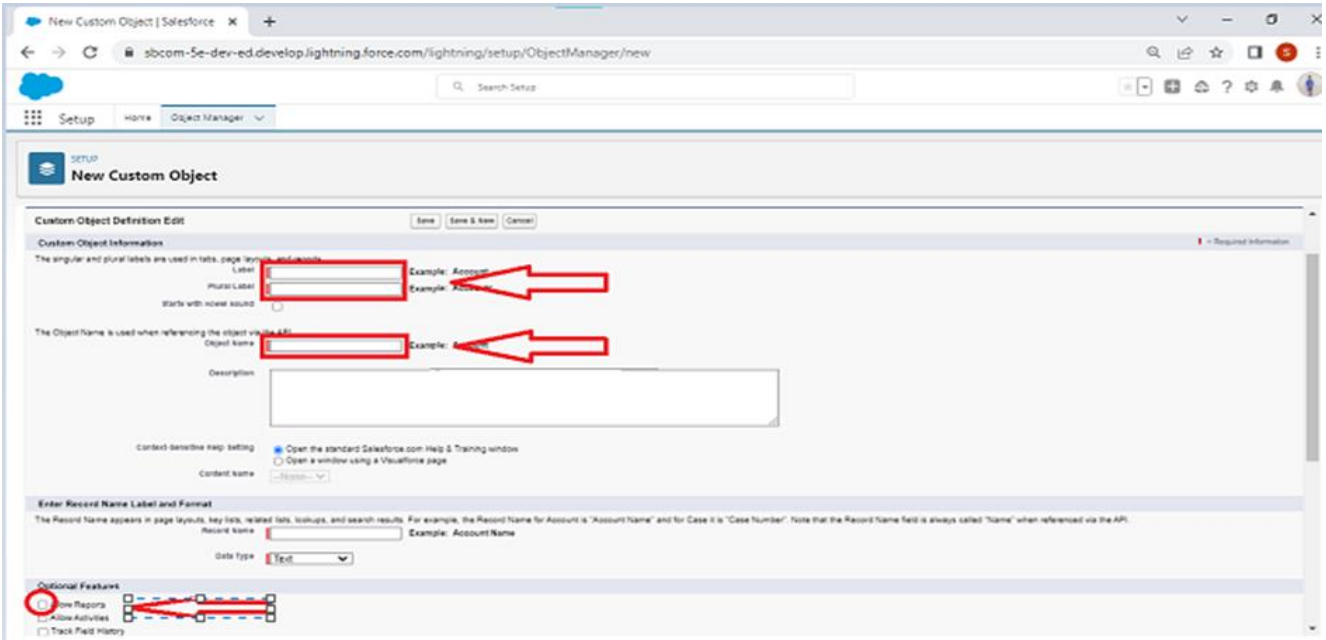
To create an object:

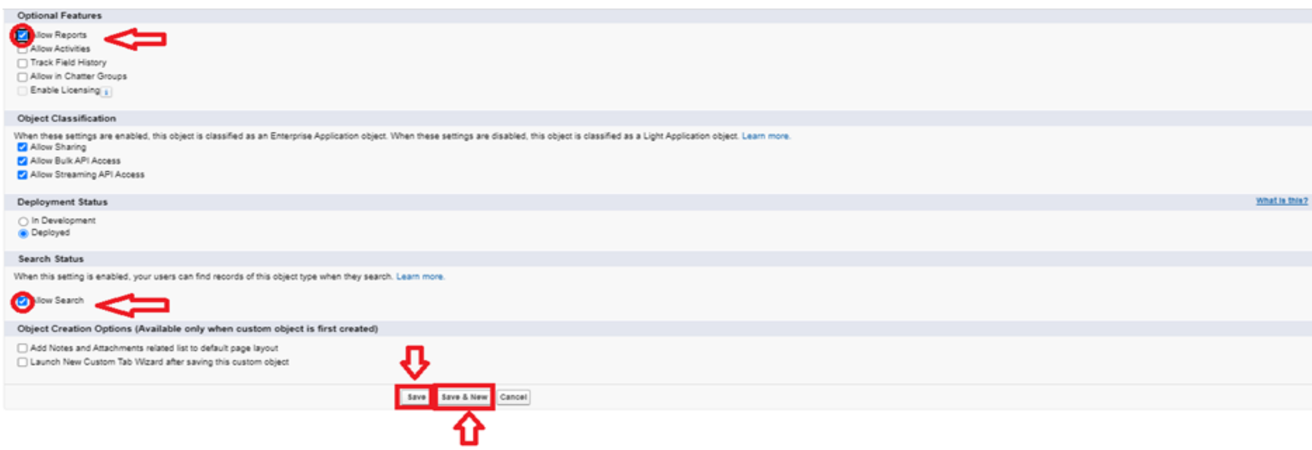
From the setup page? Click on Object Manager? Click on Create? Click on Custom Object.



On Custom object defining page:

Enter the label name, plural label name, click on Allow reports, Allow search.





Click on save

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Customer Details
2. Plural label name >> Customer Details
3. Enter Record Name Label and Format
- Record Name >> Customer Name
- Data Type >> Text
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format

- Record Name >> Appointment Name
- Data Type >> Auto Number
- Display Format >> app- {000}
- Starting number >> 1

2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format

- Record Name >> Appointment Name
- Data Type >> Auto Number
- Display Format >> app-{000}
- Starting number >> 1

2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

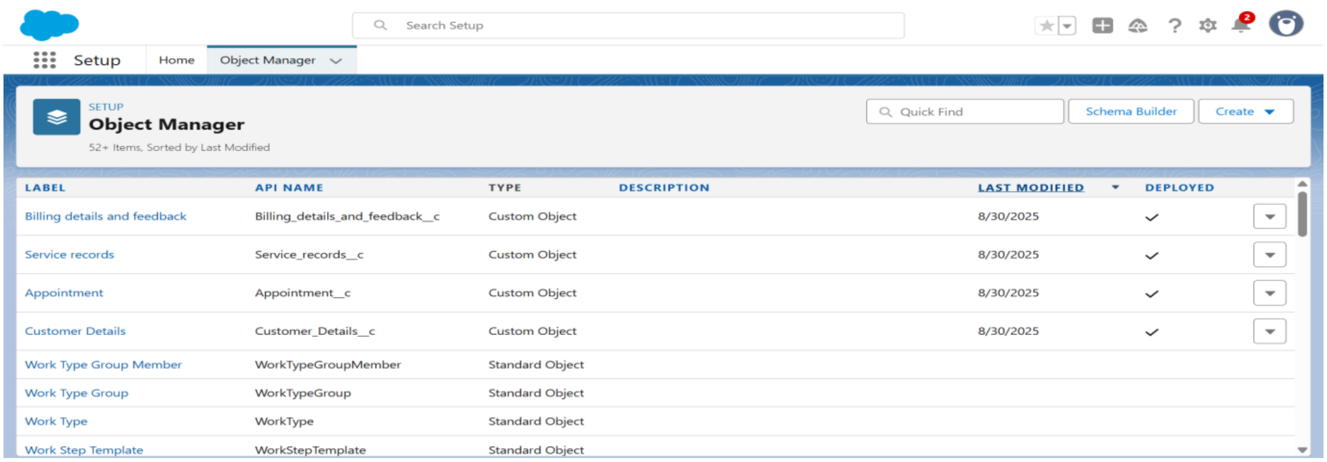
To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Billing details and feedback
2. Plural label name >> Billing details and feedback
3. Enter Record Name Label and Format

- Record Name >> Billing details and feedback Name
- Data Type >> Auto Number

- Display Format >> bill-{000}
- Starting number >> 1

2. Click on Allow reports and Track Field History,
3. Allow search >> Save.



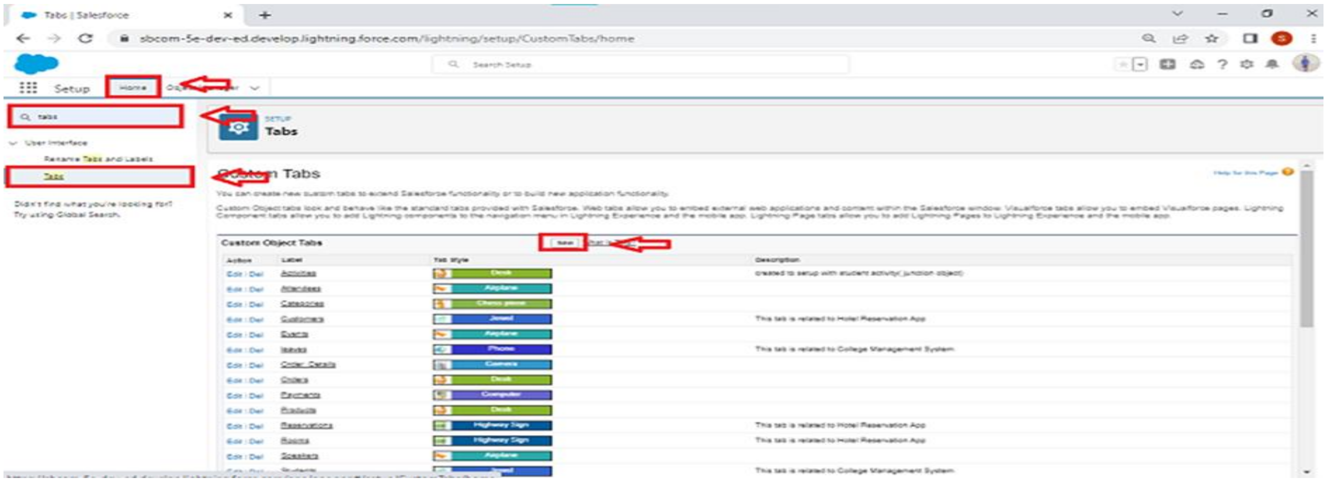
These are created Objects.

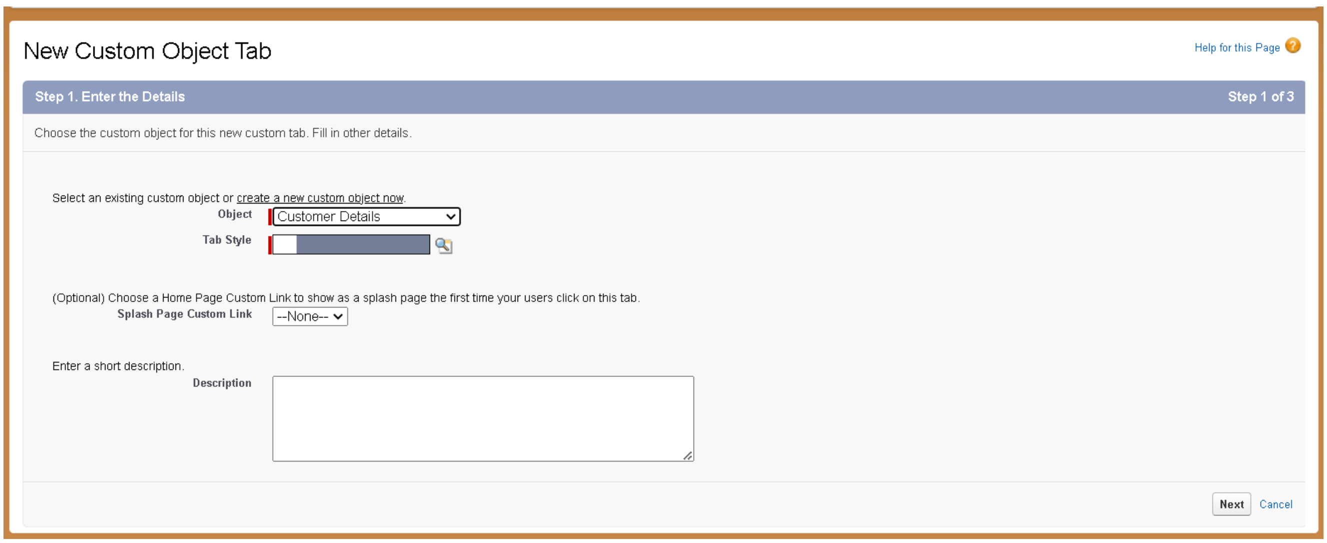
MILESTONE 3: Tabs

ACTIVITY 1: Creating custom tabs

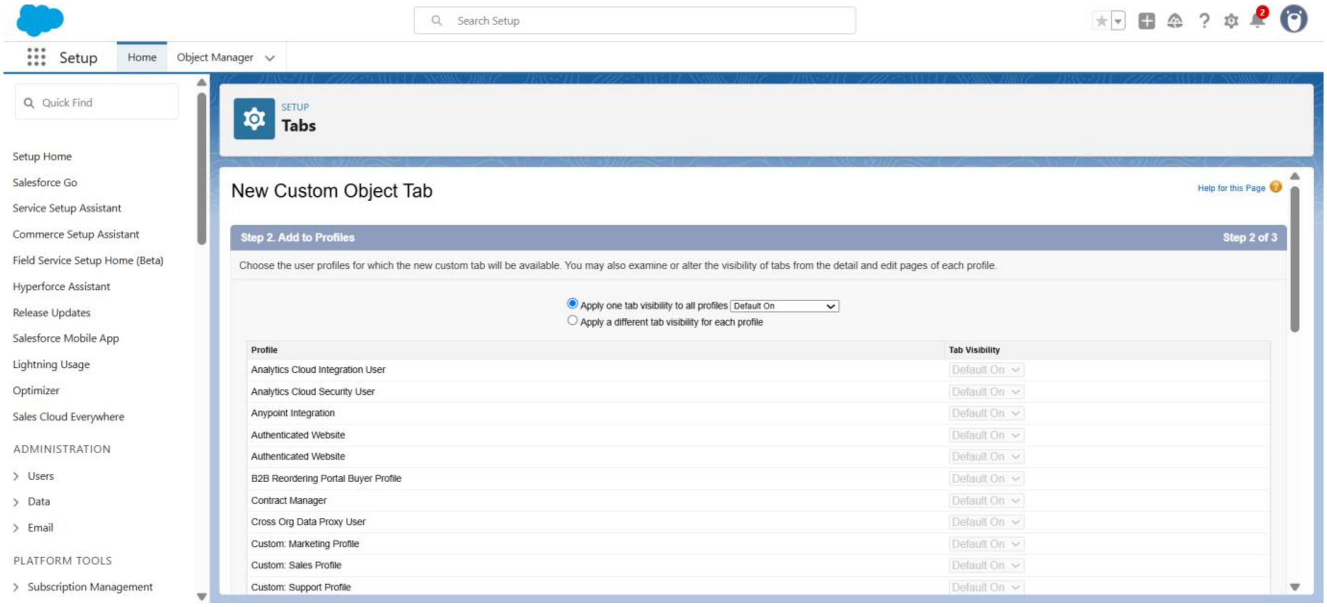
To create a Tab:(Customer Details)

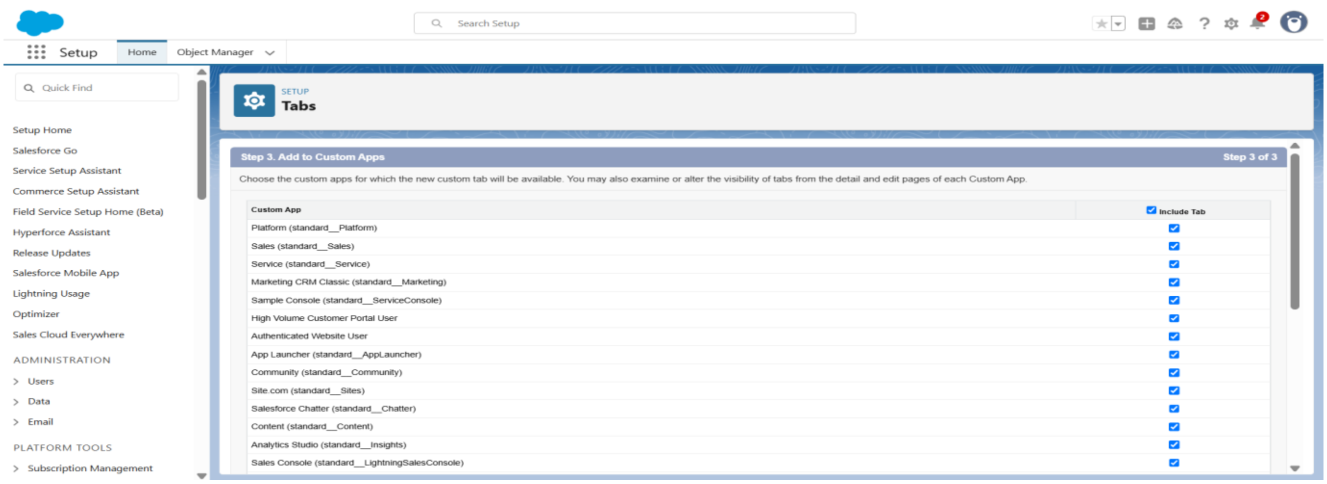
1.Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

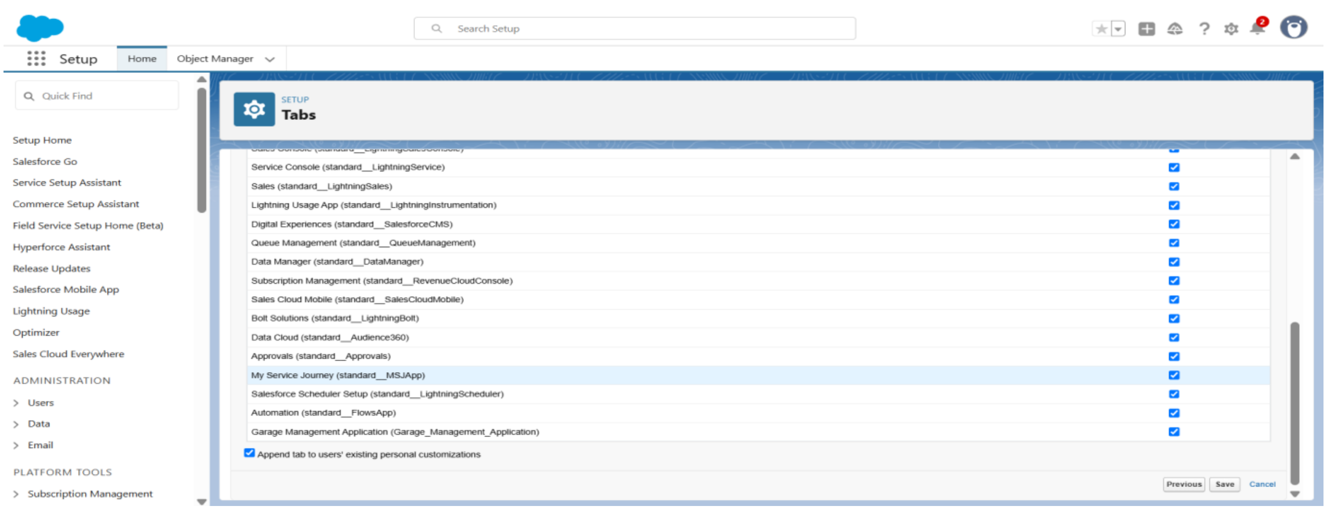

Tabs | Salesforce
Setup
Home
Tabs
Custom Tabs
Custom Object Tabs


New Custom Object Tab
Help for this Page
Step 1. Enter the Details
Step 1 of 3
Choose the custom object for this new custom tab. Fill in other details.
Select an existing custom object or create a new custom object now.
Object
Customer Details
Tab Style
(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.
Splash Page Custom Link
--None--
Enter a short description.
Description
Next
Cancel


Tab Style Selector
Create your own style
Hide styles which are used on other tabs
Airplane Alarm clock Apple Balls Bank Bell[1] Big top
Boat Books[1] Bottle Box Bridge Building Building Block
Caduceus Camera Can Car Castle CD/DVD Cell phone
Chalkboard Chess piece Chip Circle Compass Computer[1] Credit card
CRT TV Cup Desk Diamond Dice Factory Fan
Flag Form Gears Globe Guitar Hammer Hands
Handsaw Headset Heart Helicopter Hexagon Highway Sign Hot Air Balloon
Insect IP Phone Jewel Keys Laptop[1] Leaf Lightning
Locked Mail Map Measuring Tape Microphone Moon Motorcycle
Musical Note PDA Pencil People Phone Postage Presenter
Radar dish Real Estate Sign Red Cross Sack Safe Sailboat Saxophone
Scales Shield Ship Shopping Cart Square Stack of Cash Star
Stethoscope Stopwatch Street Sign Sun Telescope Thermometer Ticket
Train Treasure chest Triangle Trophy Truck TV Widescreen Umbrella
Whistle Wrench


Search Setup
Setup
Home
Object Manager
Quick Find
Setup Home
Salesforce Go
Service Setup Assistant
Commerce Setup Assistant
Field Service Setup Home (Beta)
Hyperforce Assistant
Release Updates
Salesforce Mobile App
Lightning Usage
Optimizer
Sales Cloud Everywhere
ADMINISTRATION
Users
Data
Email
PLATFORM TOOLS
Subscription Management
SETUP
Tabs
New Custom Object Tab
Help for this Page
Step 2. Add to Profiles
Step 2 of 3
Choose the user profiles for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each profile.
Apply one tab visibility to all profiles
Default On
Apply a different tab visibility for each profile
Profile
Tab Visibility
Analytics Cloud Integration User
Analytics Cloud Security User
Anypoint Integration
Authenticated Website
Authenticated Website
B2B Reordering Portal Buyer Profile
Contract Manager
Cross Org Data Proxy User
Custom: Marketing Profile
Custom: Sales Profile
Custom: Support Profile


Search Setup
Setup
Home
Object Manager
Quick Find
SETUP
Tabs
Step 3. Add to Custom Apps
Step 3 of 3
Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.
Custom App
Include Tab
Platform (standard__Platform)
Sales (standard__Sales)
Service (standard__Service)
Marketing CRM Classic (standard__Marketing)
Sample Console (standard__ServiceConsole)
High Volume Customer Portal User
Authenticated Website User
App Launcher (standard__AppLauncher)
Community (standard__Community)
Site.com (standard__Sites)
Salesforce Chatter (standard__Chatter)
Content (standard__Content)
Analytics Studio (standard__Insights)
Sales Console (standard__LightningSalesConsole)


Search Setup
Setup
Home
Object Manager
Quick Find
SETUP
Tabs
Service Console (standard__LightningService)
Sales (standard__LightningSales)
Lightning Usage App (standard__LightningInstrumentation)
Digital Experiences (standard__SalesforceCMS)
Queue Management (standard__QueueManagement)
Data Manager (standard__DataManager)
Subscription Management (standard__RevenueCloudConsole)
Sales Cloud Mobile (standard__SalesCloudMobile)
Bolt Solutions (standard__LightningBolt)
Data Cloud (standard__Audience360)
Approvals (standard__Approvals)
My Service Journey (standard__MSJApp)
Salesforce Scheduler Setup (standard__LightningScheduler)
Automation (standard__FlowsApp)
Garage Management Application (Garage_Management_Application)
Append tab to users' existing personal customizations
Previous
Save
Cancel

ACTIVITY 2: Creating remaining tabs

1.Now create the Tabs for the remaining Objects, they are “ Appointments, Service records,Billing details and feedback”.

2.Follow the same steps as mentioned in Activity -1.

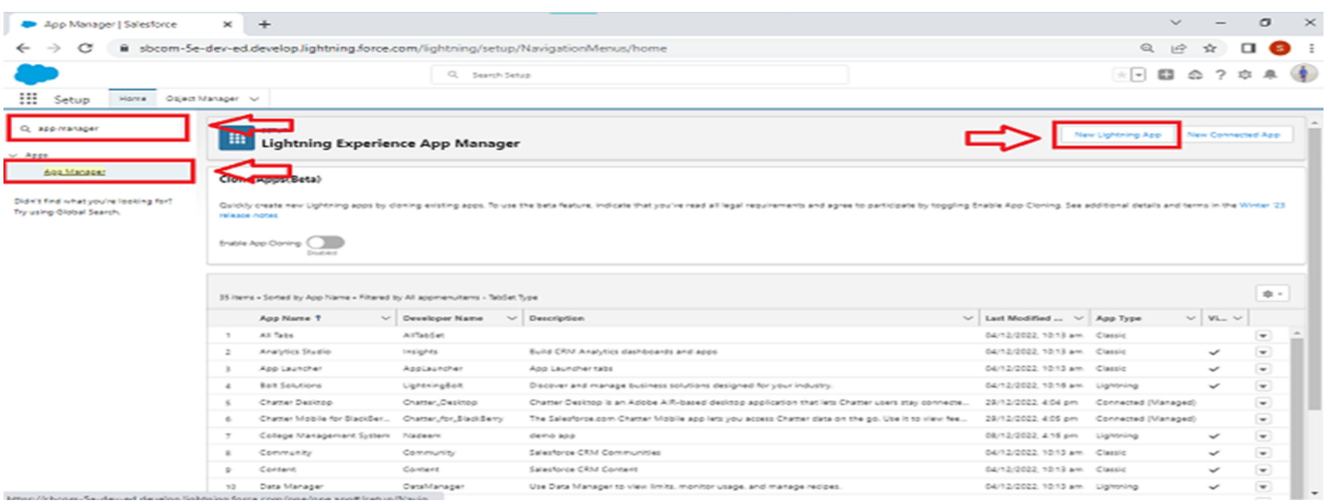
Finally new custom tabs are created Successfully

MILESTONE 4: The Lightning App

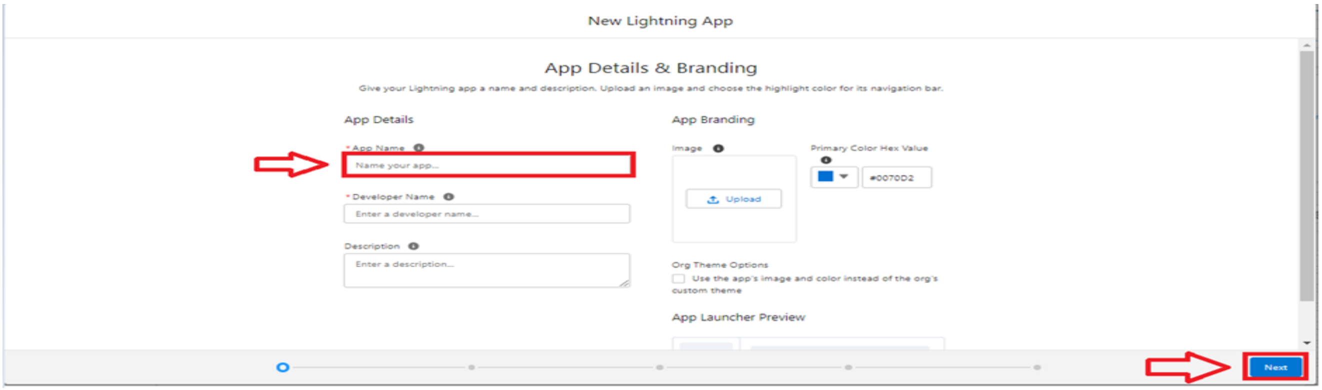
ACTIVITY 1: Creating a Lightning App

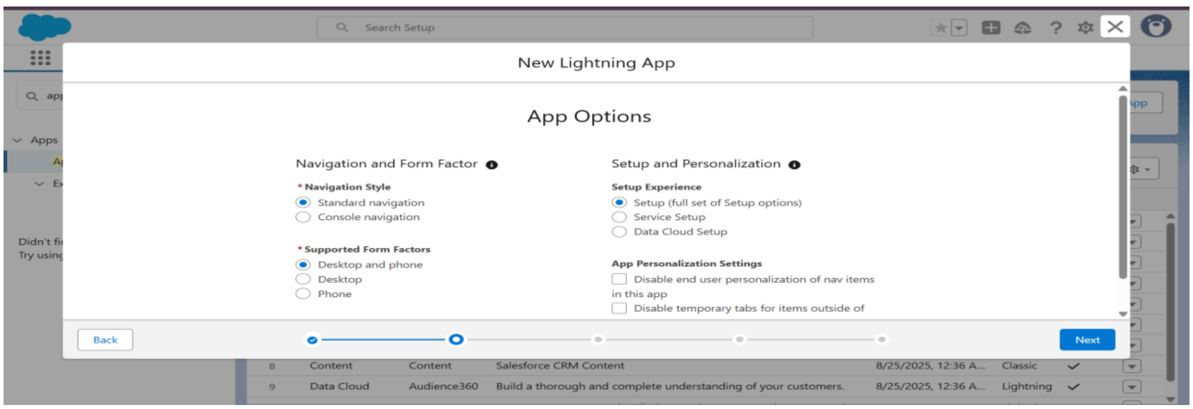
To create a lightning app page:

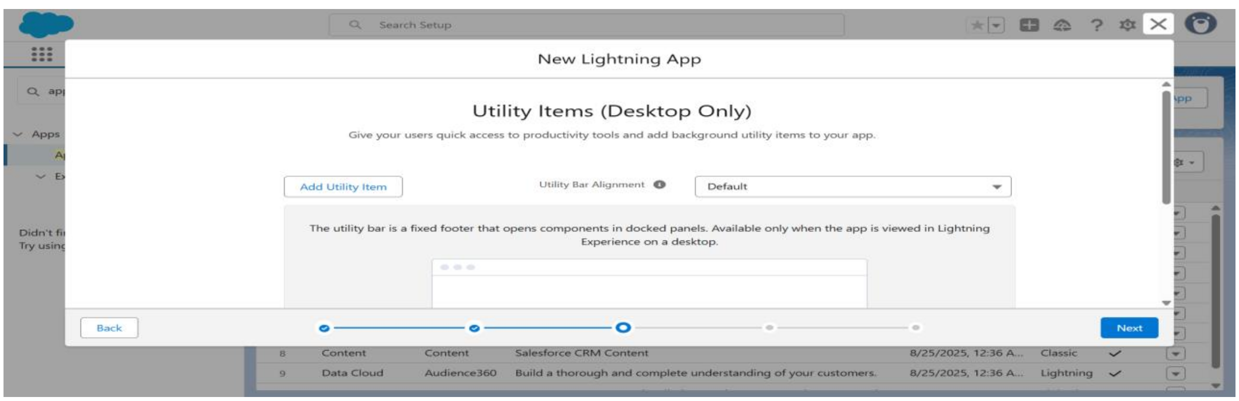
1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on new lightning App.



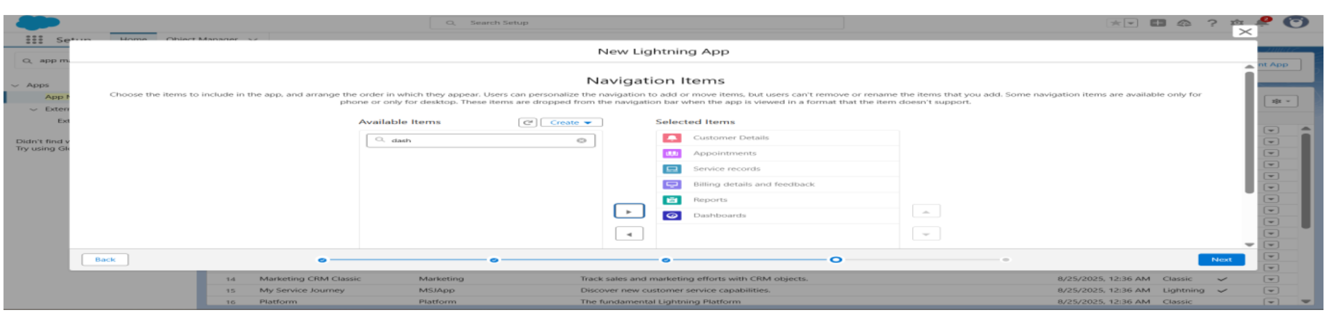
2.Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.





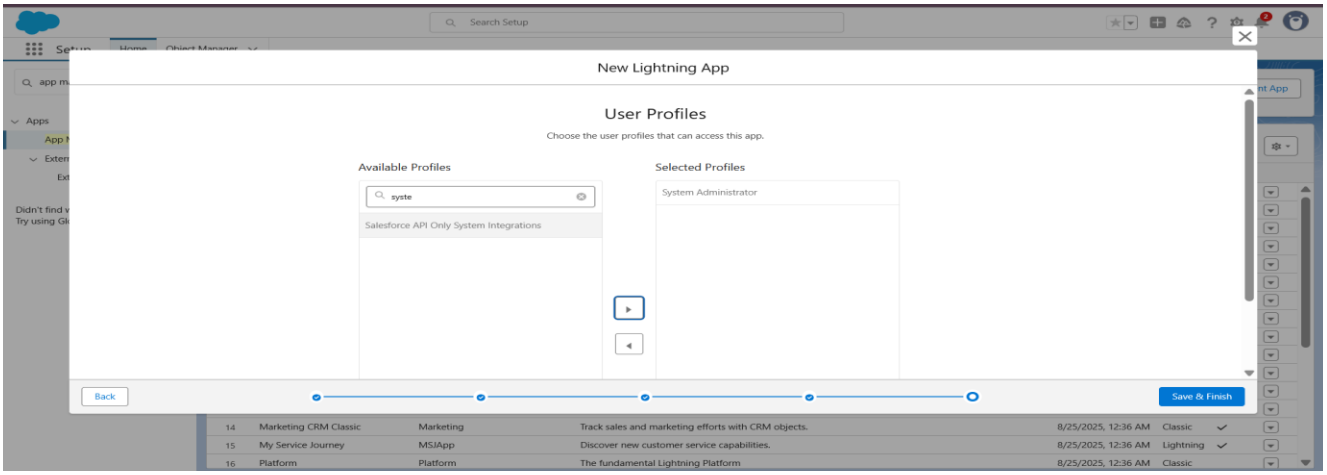


3.To Add Navigation Items:

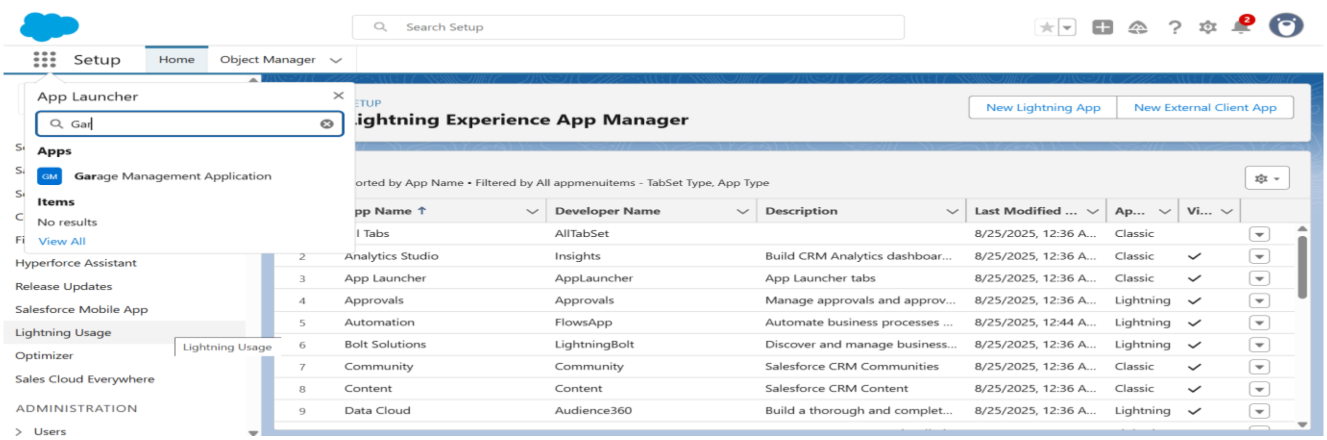


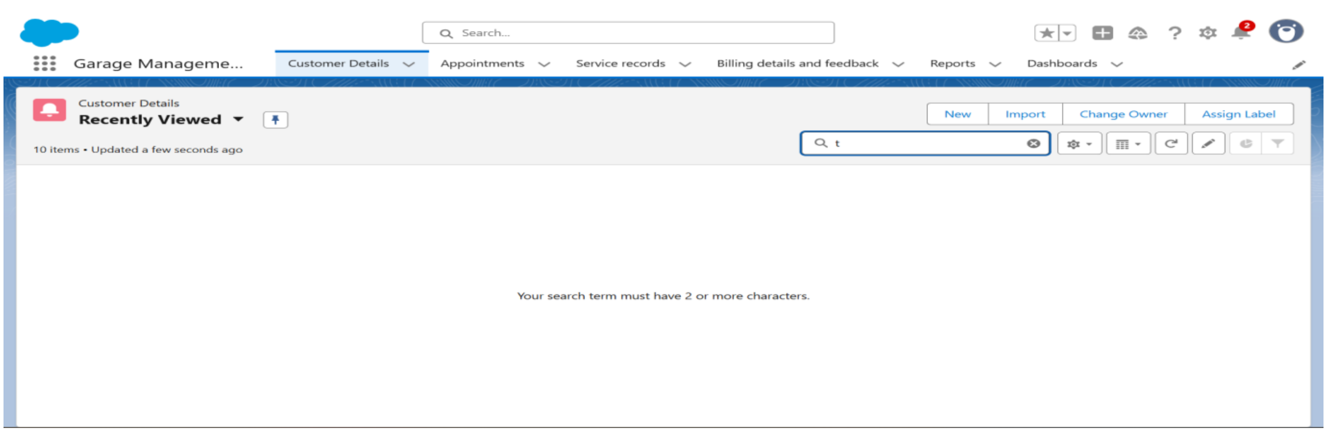
4.Select the items (Customer Details,Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.

5.To Add User Profiles :



Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.





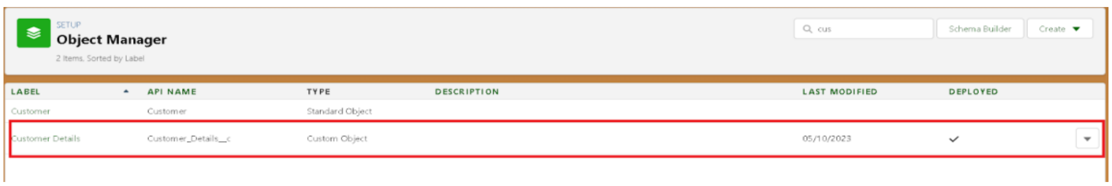
New Lightning app was created successfully.

MILESTONE 5: Fields

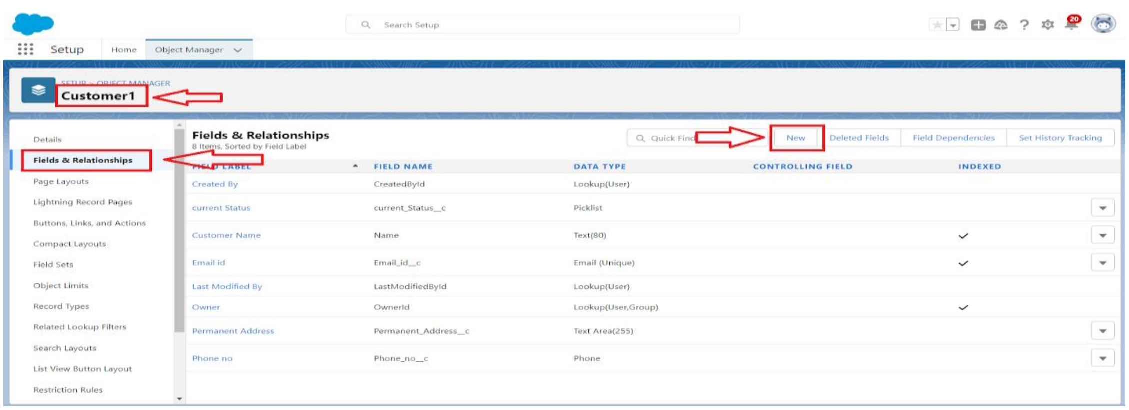
ACTIVITY 1: Creation of fields for the Customer Details object

1. To create fields in an object:

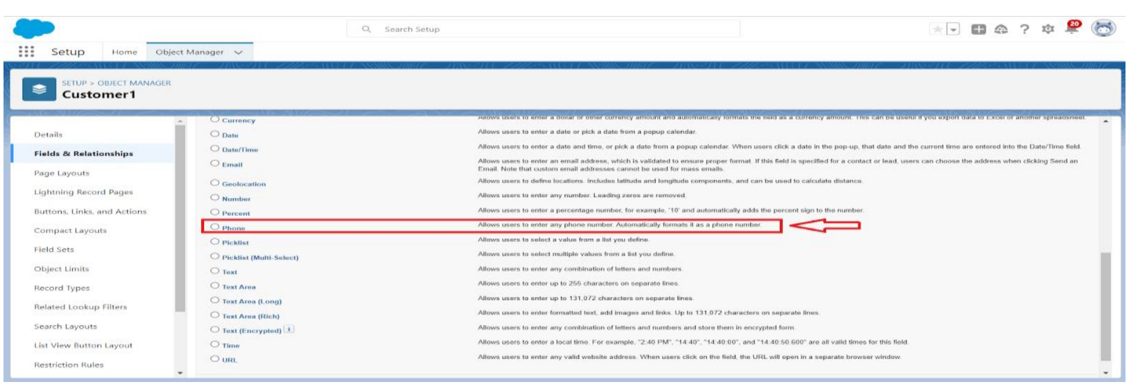
1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.



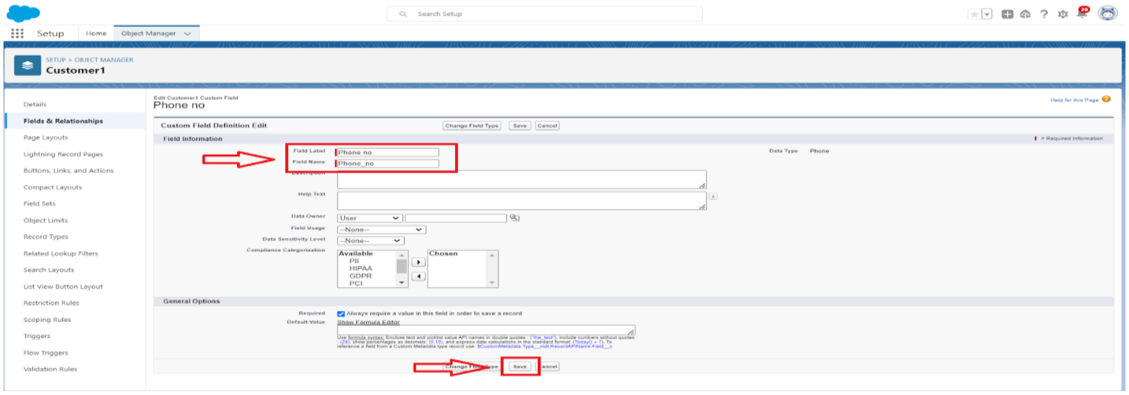
2. Now click on "Fields & Relationships" >> New



3. Select Data Type as a "Phone"



4. Click on next.



5. Fill the Above as following:

- Field Label: Phone number
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

2. To create other fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New
3. Select Data type as a "Email" and Click on Next
4. Fill the Above as following:

- Field Label: Gmail
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

ACTIVITY 2: Creation of Lookup Fields

1.Creation of Lookup Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New
3. Select "Look-up relationship" as data type and click Next.
4. Select the related object " Customer Details" and click next.
5. Next >> Next >> Save.

Note: Make sure you complete Activity 4 Before continuing.

2.Creation of Lookup Field on Service records Object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Appointment ” and click next.
5. Make it a required field so click on Required.
6. Scroll down for Lookup Filter and click on Show filter settings.
7. Now add the filter criteria.
8. Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date
9. Filter type should be Required.
10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.
12. Next >> Next >> Save.

3.Creation of Lookup Field on Billing details and feedback Object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Service records” and click next.
5. Next >> Next >> Save & new.

ACTIVITY 3: Creation of Checkbox Fields

1. Creation of Checkbox Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.

3. Select “Check box” as data type and click Next.
4. Give the Field Label: Maintenance service
5. Field Name: is auto populated
6. Default value: unchecked
7. 7.Click on next >> next >> save.

2.Creation of Another Checkbox Field on Appointment Object :

1. Repeat the steps from 1 to 3.
2. Give the Field Label: Repairs
3. Field Name: is auto populated
4. Default value: unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkbox with given names
7. Give the Field Label: Replacement Parts
8. Field Name: is auto populated
9. Default value: unchecked
10. Click on next >> next >> save.

3.Creation of Checkbox Field on Service records Object:

1. Go to setup >> click on Object Manager >> type object name (Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.
4. Give the Field Label: Quality Check Status
5. Field Name: is auto populated
6. Default value: unchecked
7. Click on next >> next >> save.

ACTIVITY 4: Creation of Date Fields

1.Creation of Date Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Date” as data type and click Next.

4. Give the Field Label: Appointment Date
5. Field Nme : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >> next >> save.

ACTIVITY 5: Creation of Currency Fields

1.Creation of Currency Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Currency” as data type and click Next.
4. Give the Field Label: Service Amount
5. Field Name: is auto populated
6. Click on next
7. Give read only for all the profiles in field level security for profile.
8. Click on next > > save.

2.Creation of Currency Field on Billing details and feedback Object

Follow the same steps as mentioned above in Billing details and feedback Object

1. Change the label name as mentioned.
2. Give the Field Label: Payment Paid
3. Field Name : is auto populated

ACTIVITY 6: Creation of Text Fields

1.Creation of Text Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Text” as data type and click Next.
4. Give the Field Label : Vehicle number plate
5. Field Name : is auto populated
6. Length : 10

7. Make field as Required and Unique.
8. Click on next >> next >> save.

2.Creation of Text Fields in Billing details and feedback object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “text” as data type and click Next.
4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required.
8. Click on next >> next >> save

ACTIVITY 7: Creation of Picklist Fields

1.Creation of Picklist Fields in Service records object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Service Status”, under values select “Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are: Started, Completed.
6. Click Next.
7. Next >> Next >> Save.

2. Creation of Picklist Fields in Billing details and feedback object :

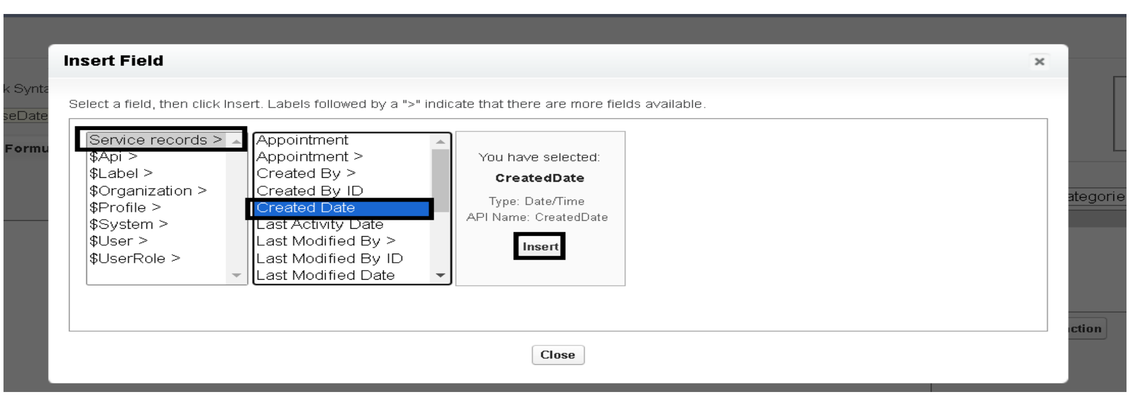
1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.

4. Enter Field Label as “Payment Status”, under values select “Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are: Pending, Completed.
6. Click Next.
7. Next >> Next >> Save.

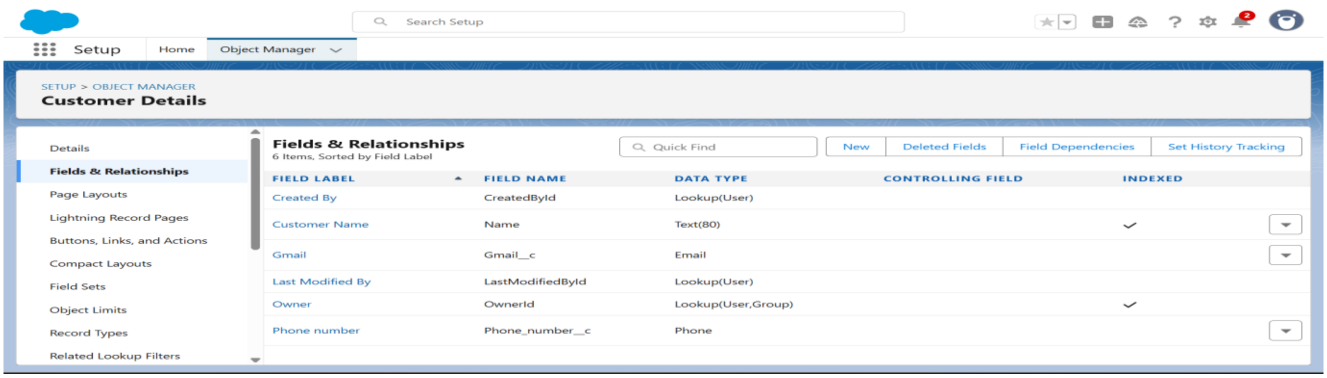
ACTIVITY 8: Creation of Formula Field in Service records Object

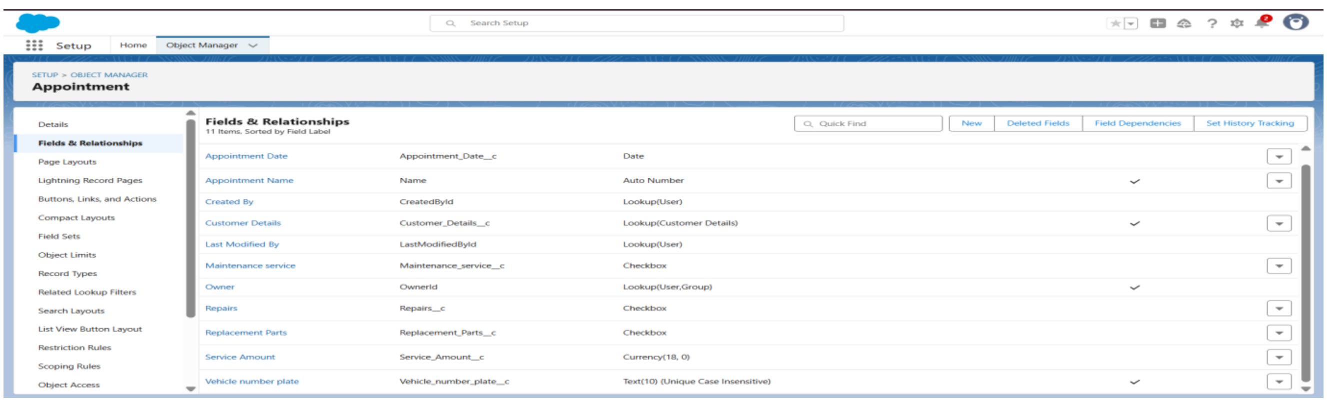
1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “service date” and select formula return type as “Date” and click next.

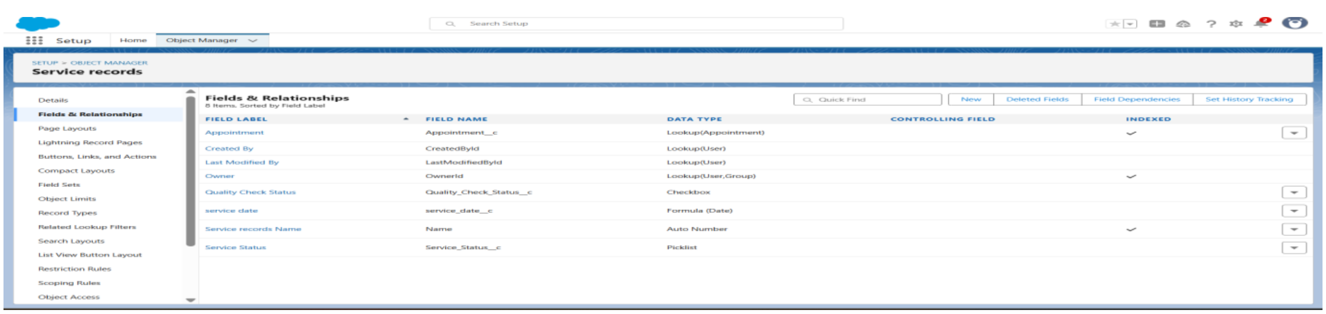
5.Insert field formula should be : CreatedDate



6. click "Check Syntax" .
7. Click next >> next >> Save.





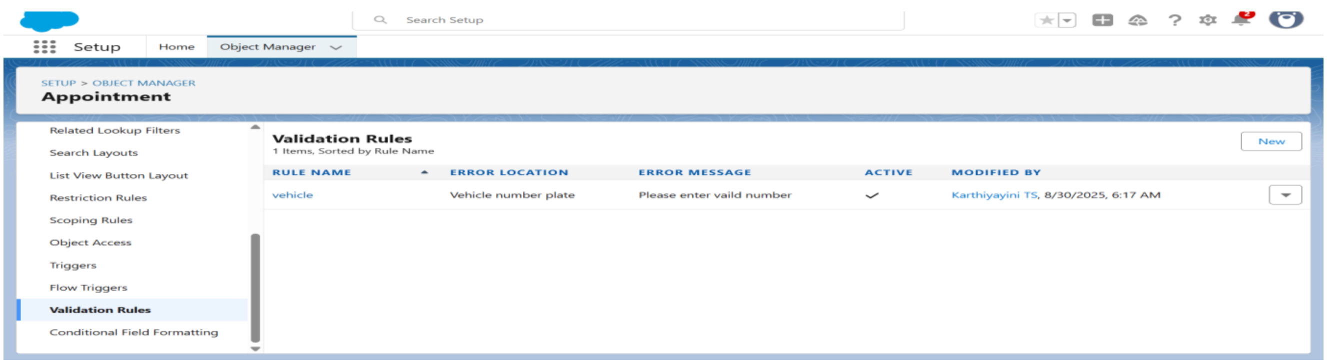


NOTE: Thus we have successfully created all the fields which instructed to create on the respective object .

MILESTONE 6: Validation Rule

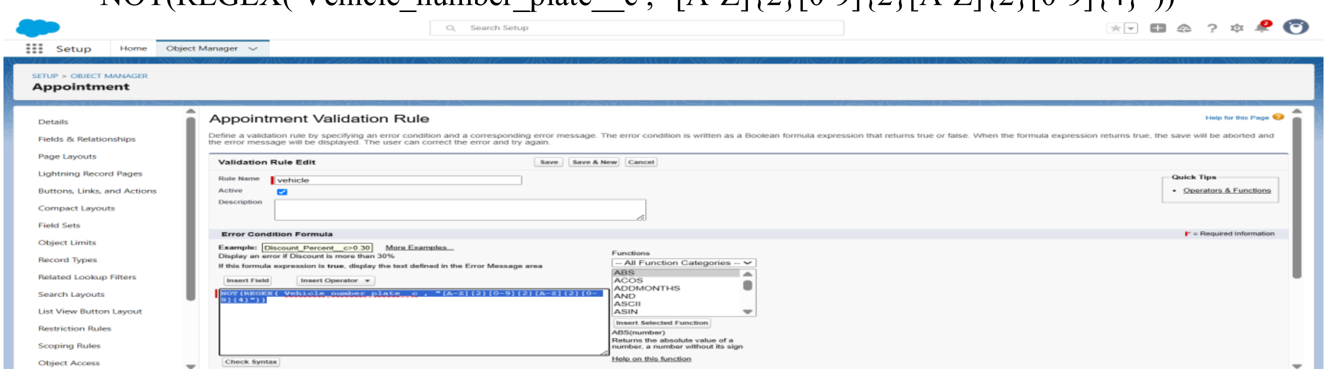
ACTIVITY 1: To create a validation rule to an Appointment Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.

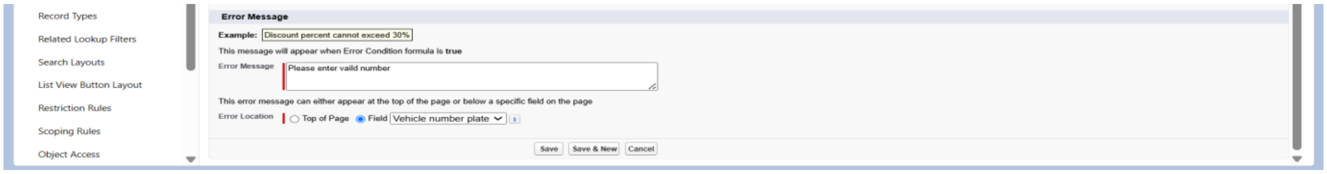


3. Enter the Rule name as " Vehicle ".
4. Insert the Error Condition Formula as : -

NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))



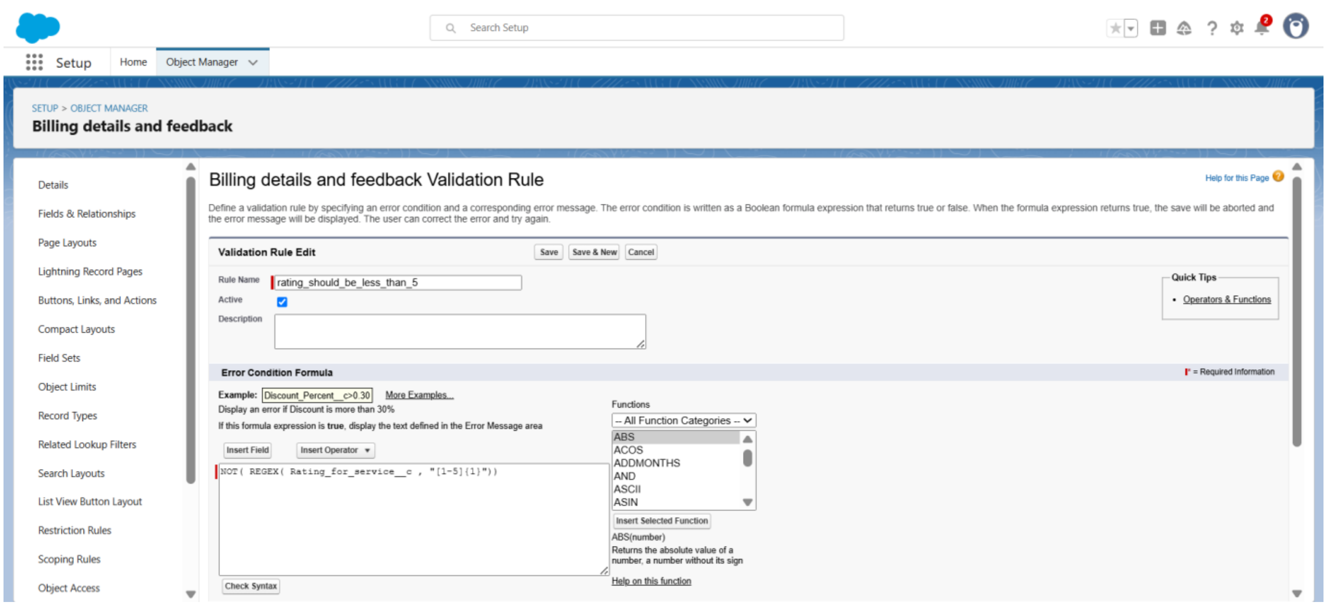
5. Enter the Error Message as "Please enter vaild number ", select the Error location as Field and select the field as "Vehicle number plate", and click Save.



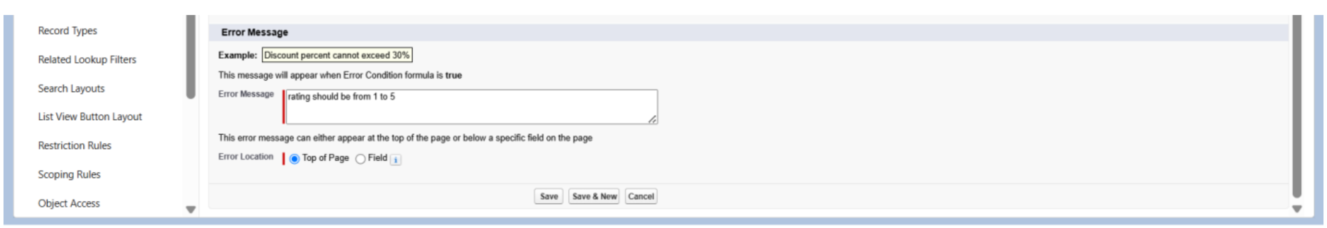
ACTIVITY 2:To create a validation rule to an Billing details & feedback Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " rating_should_be_less_than_5".
4. Insert the Error Condition Formula as : -

NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))



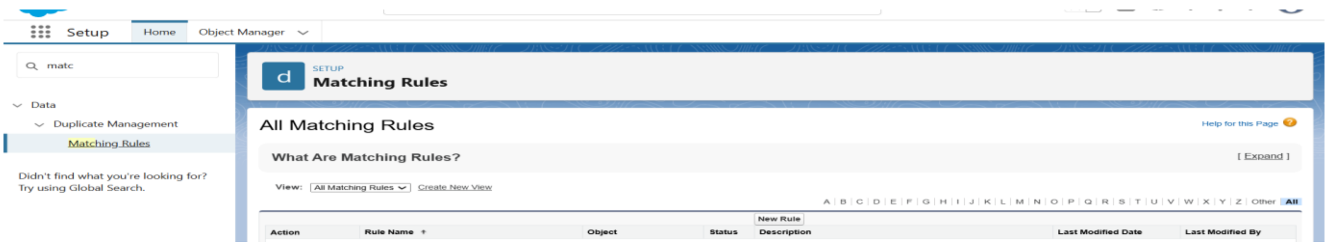
5. Enter the Error Message as "rating should be from 1 to 5", select the Error location as Field and select the field as "Rating for Service", and click Save.



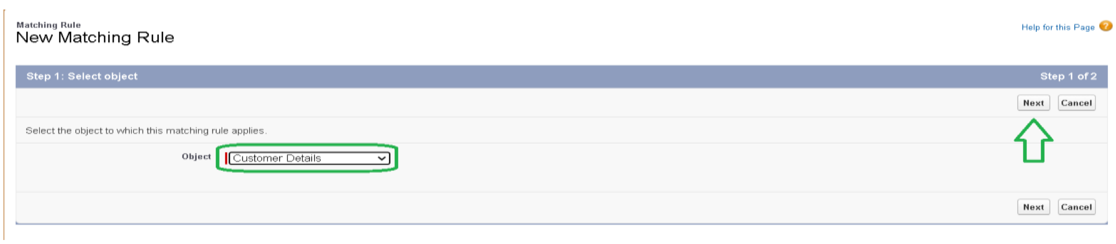
MILESTONE 7: Duplicate Rule

ACTIVITY 1: To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.



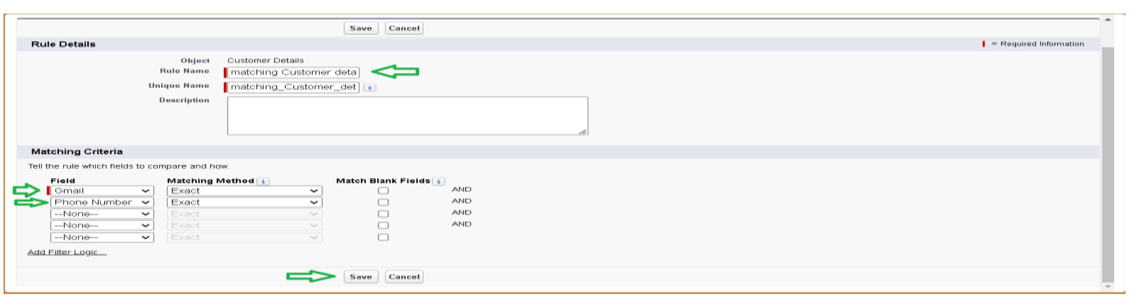
3. Select the object as Customer details and click Next.



4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method

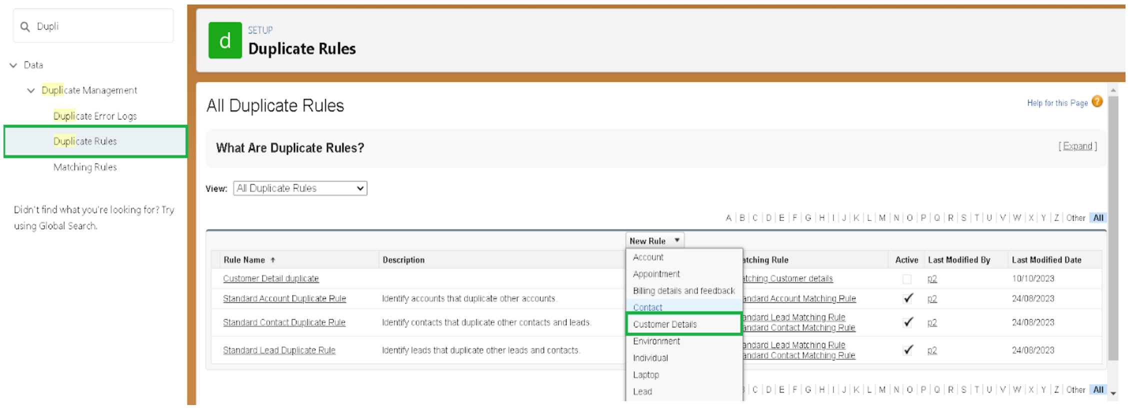
Field	Matching Method
1. Gmail	Exact
2. Phone Number	Exact

8. Click save.
9. After Saving Click on Activate.

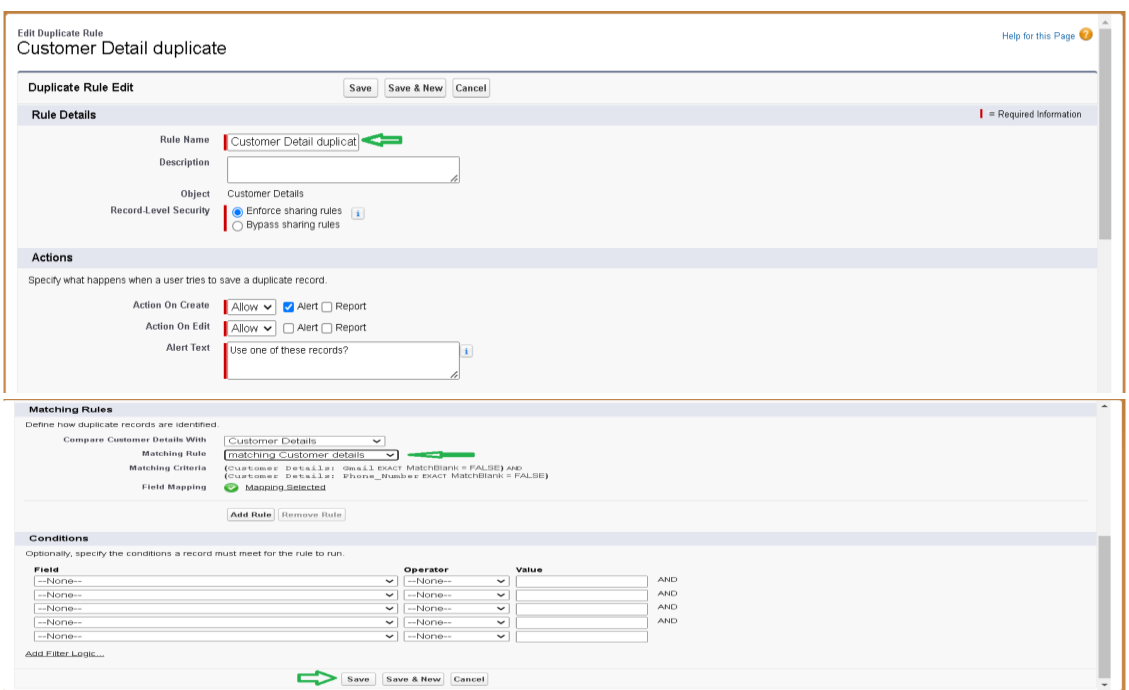


ACTIVITY 2: To create a duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.



3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

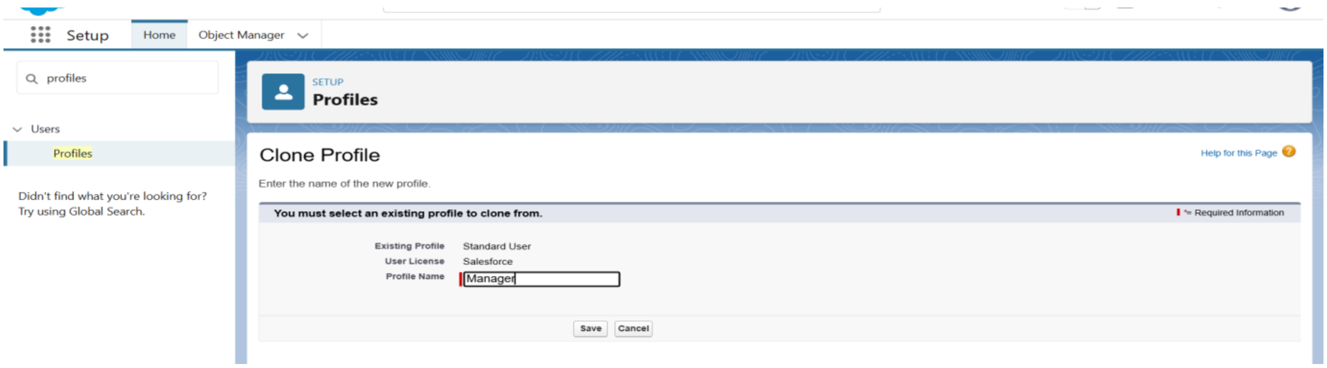


MILESTONE 8: Profiles

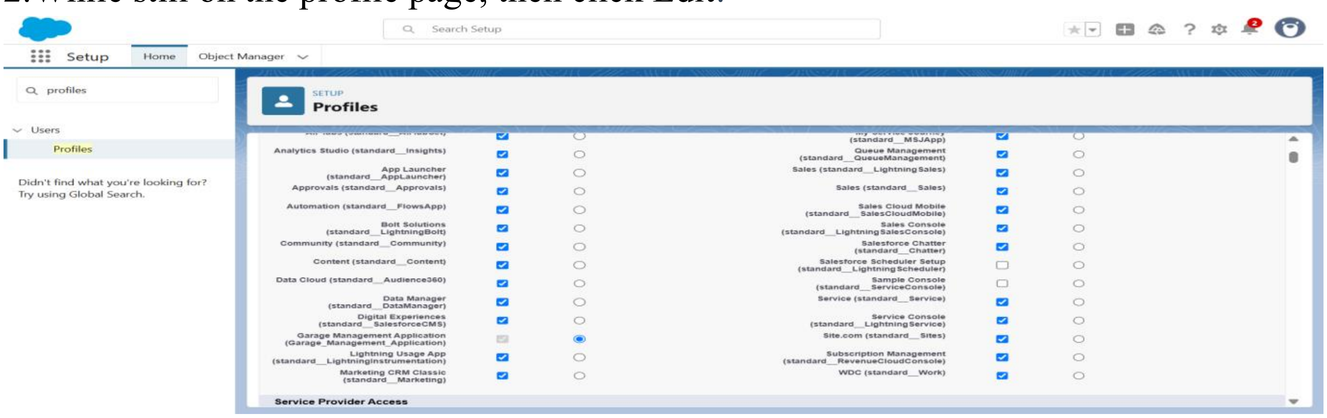
ACTIVITY 1: Manager Profile

1. To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.

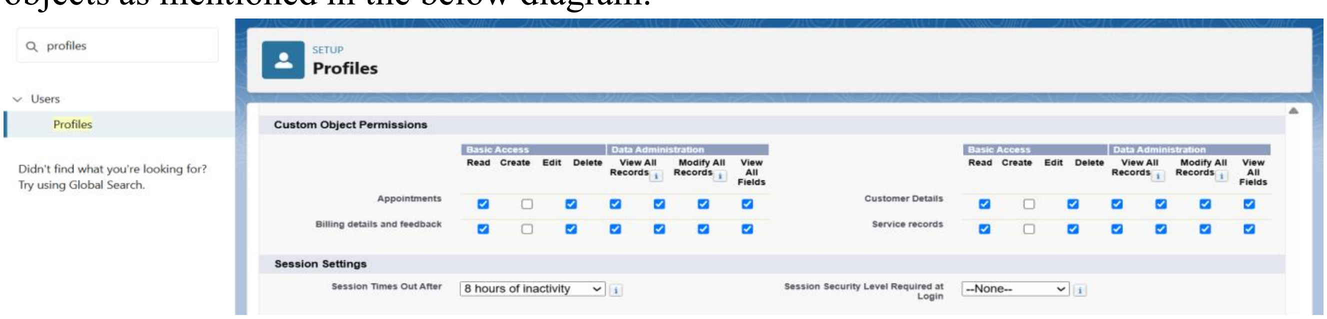


2.While still on the profile page, then click Edit.



3.Select the Custom App settings as default for the Garage management.

4.Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.

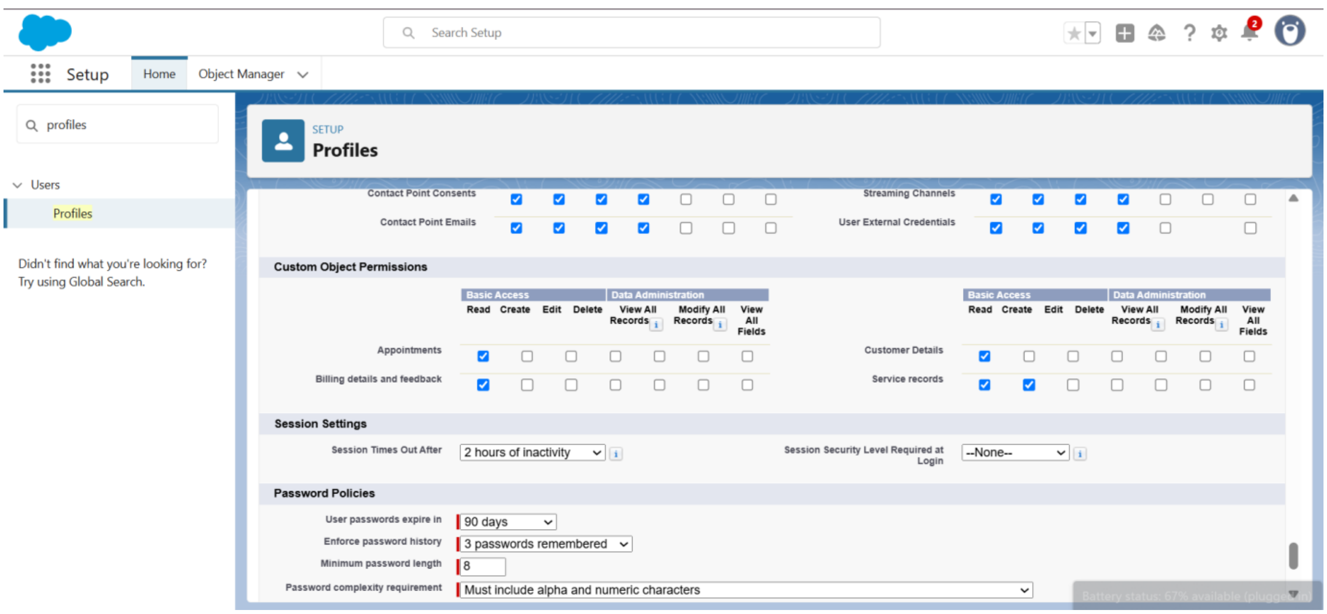


5. Changing the session times out after should be “ 8 hours of inactivity”.

6. Change the password policies as mentioned :
7. User passwords expire in should be “ never expires ”.
8. Minimum password length should be “ 8 ”, and click save.

ACTIVITY 1: Sales person Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Salesforce Platform User) >> enter profile name (sales person) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the GArage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.



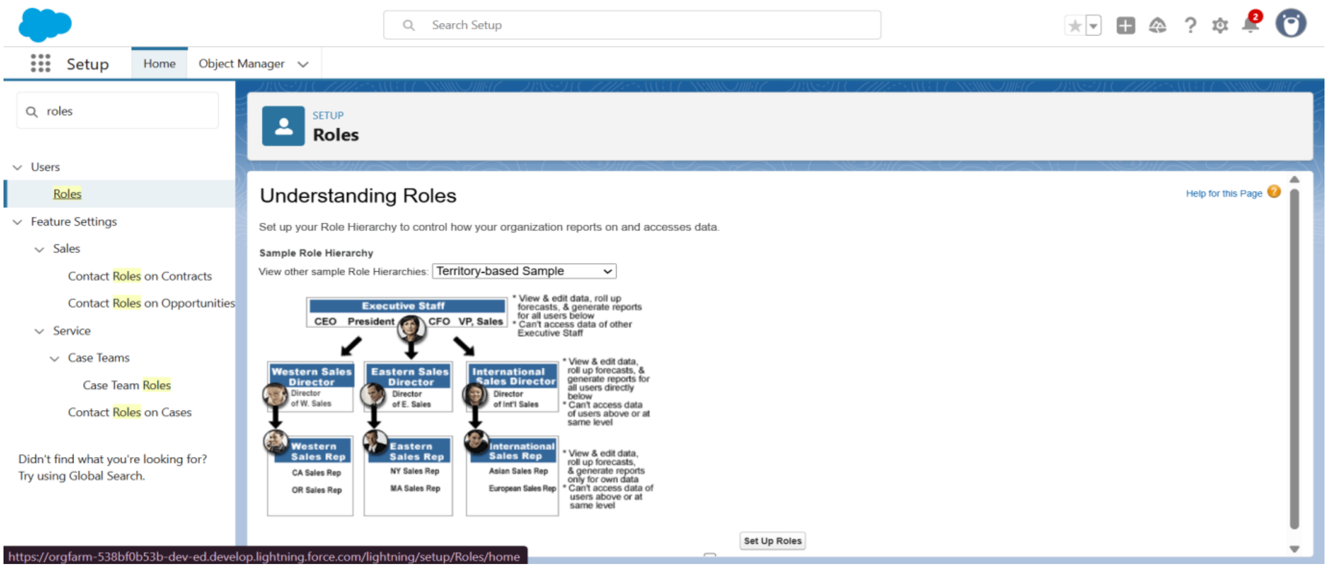
5. And click save.

MILESTONE 9: Role & Role Hierarchy

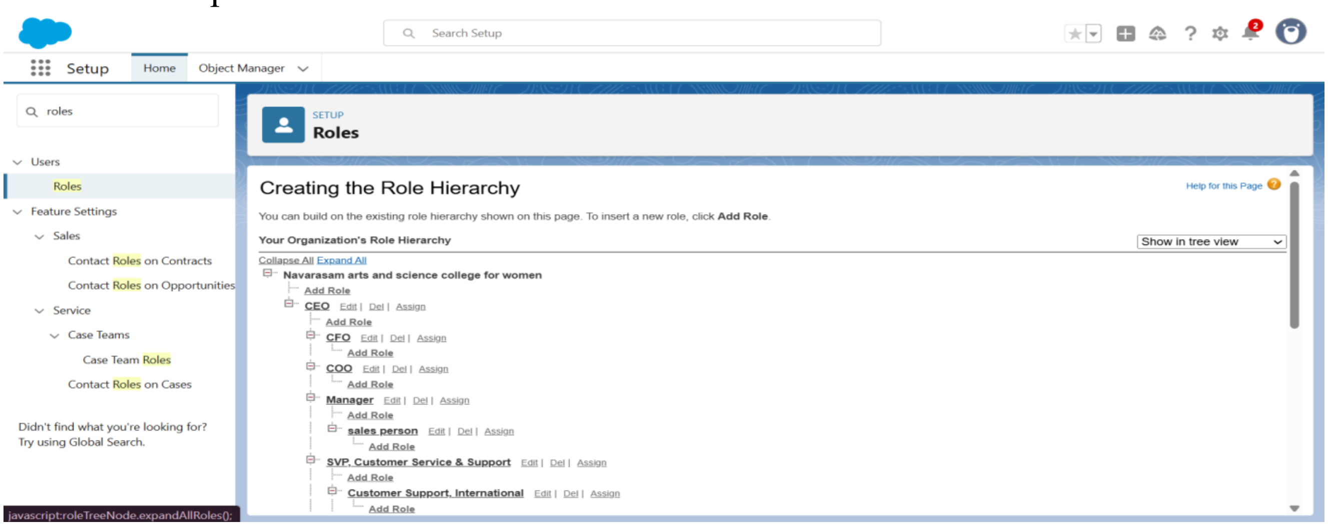
ACTIVITY 1: Creating manager role

Creating Manager Role:

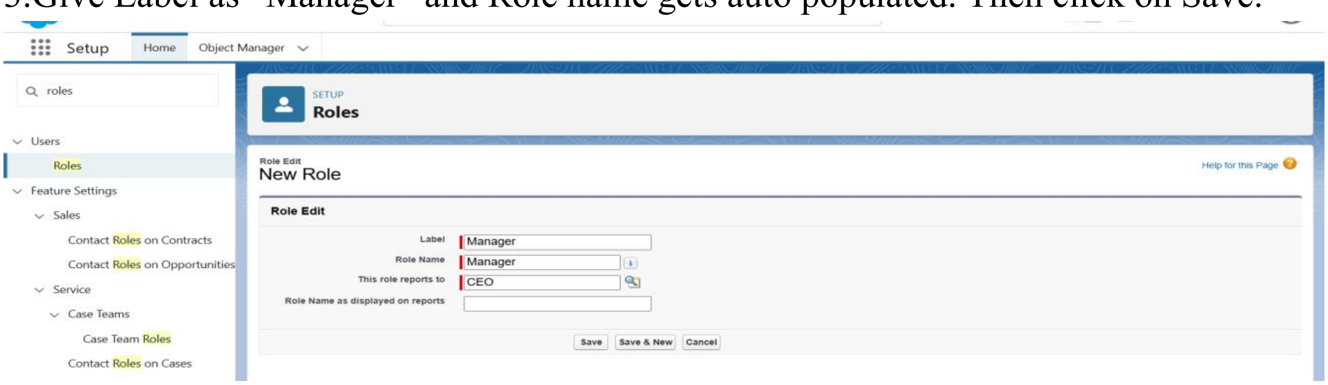
1. Go to quick find >> Search for Roles >> click on set up roles.



2.Click on Expand All and click on add role under whom this role works.



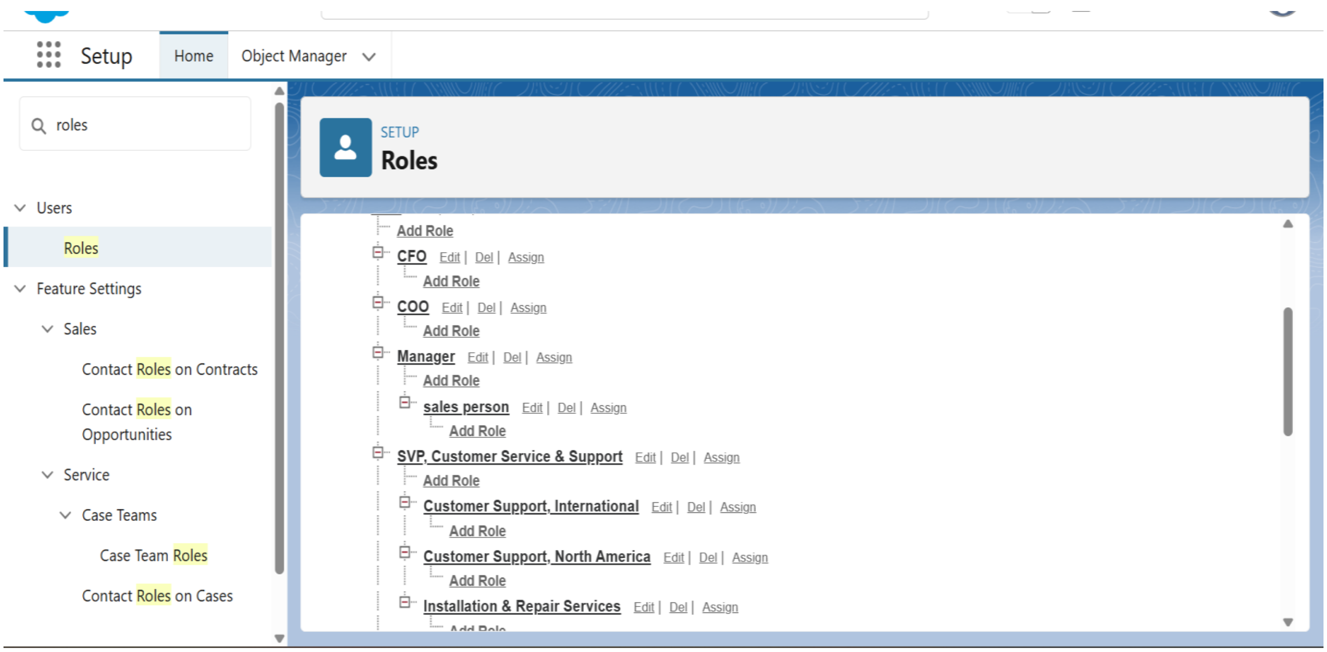
3.Give Label as “Manager” and Role name gets auto populated. Then click on Save.



ACTIVITY 2: Creating another role

Creating another two roles under manager

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.



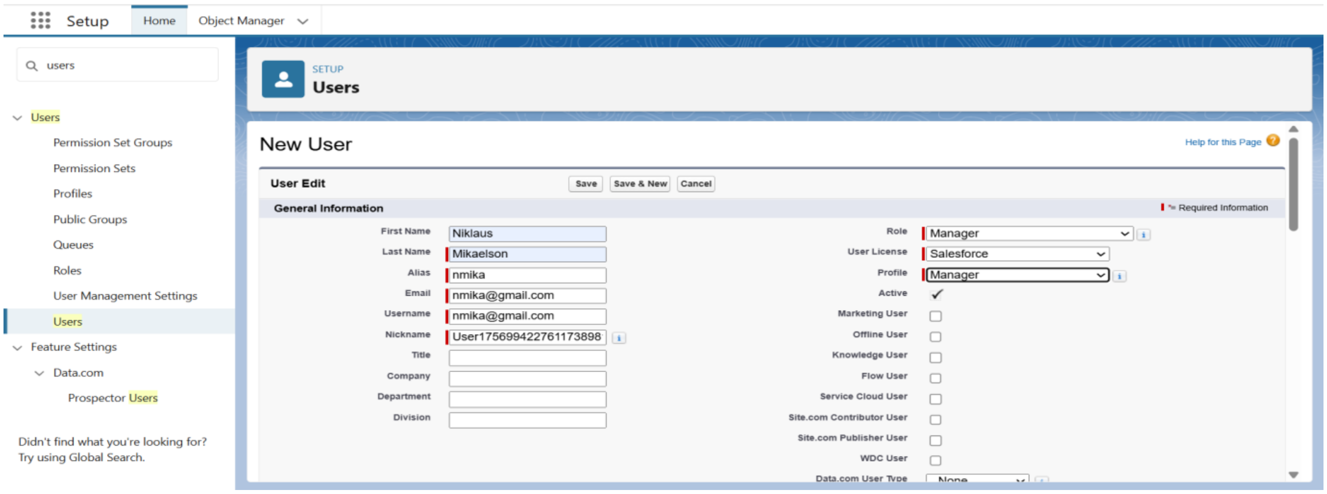
3. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

MILESTONE 10: Users

ACTIVITY 1: Create a user

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
1. First Name : Niklaus
2. Last Name : Mikaelson
3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role : Manager

8. User licence : Salesforce
9. Profiles : Manager
10. Click on save



ACTIVITY 2: Create another user

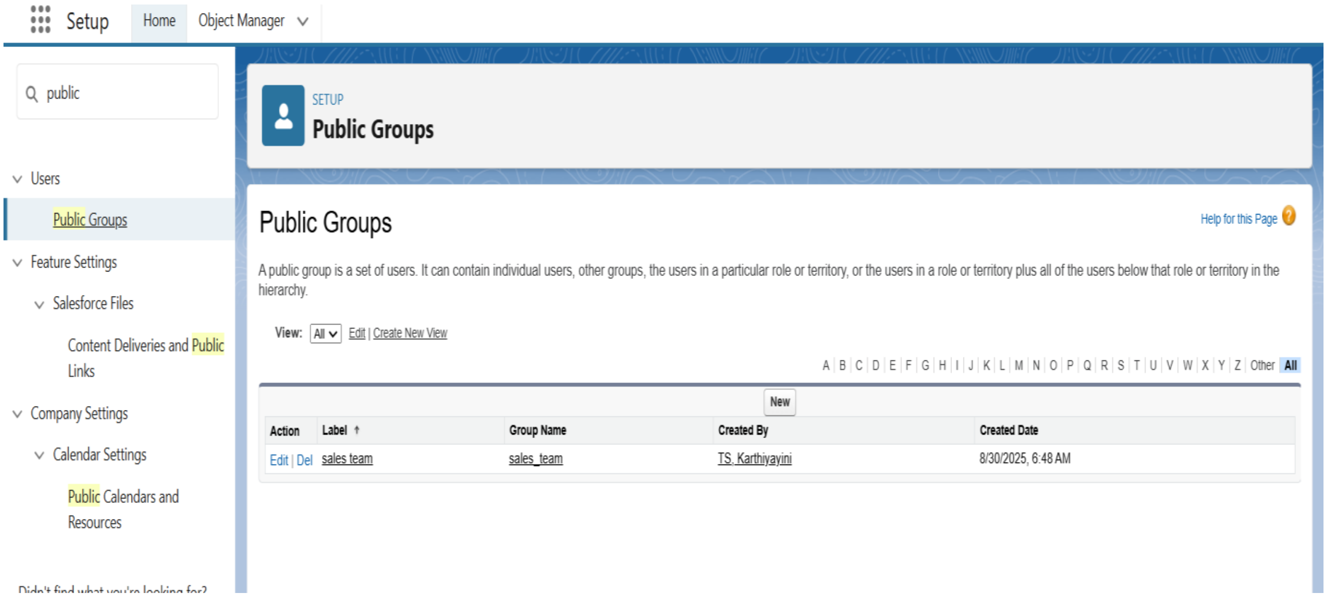
1. Repeat the steps and create another user using
    a. Role : sales person
    b. User licence : Salesforce Platform
    c. Profile : sales person

Note : create atleast 3 users with these permissions.

MILESTONE 11: Public Groups

ACTIVITY 1:Creat a new public group

1. Go to setup >> type users in quick find box >> select public groups >> click New.

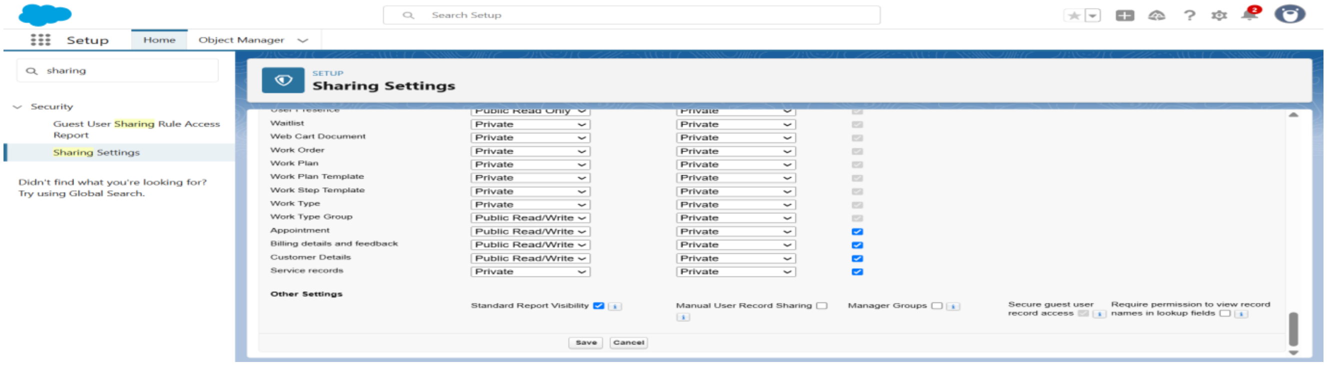


2. Give the Label as “sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.

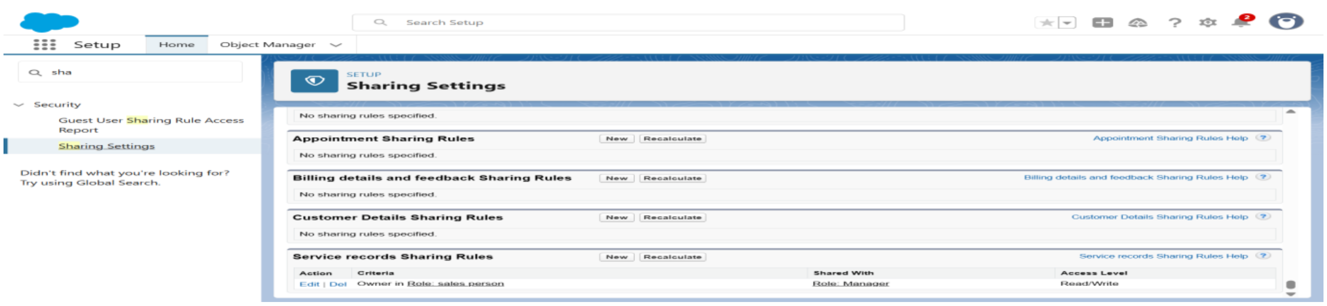
MILESTONE 12: Sharing Settings

ACTIVITY 1: Creating a sharing setting

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.



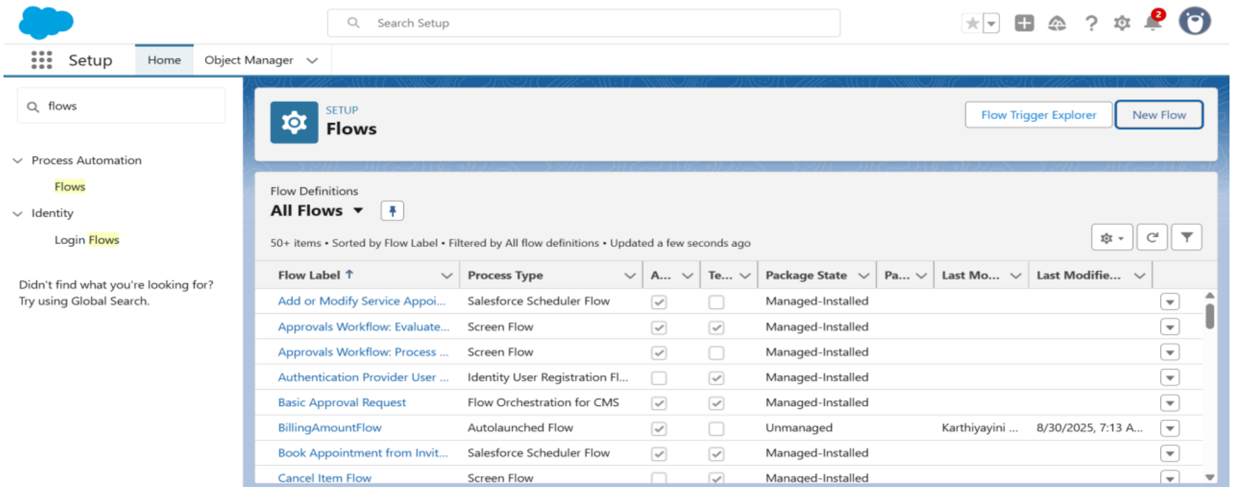
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
5. Give the Label name as " Sharing setting"
6. Rule name is auto populated.
7. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
8. In step 4: share with, select " Roles " >> " Manager "
9. In step 5 : Change the access level to " Read / write ".
10. Click on save.



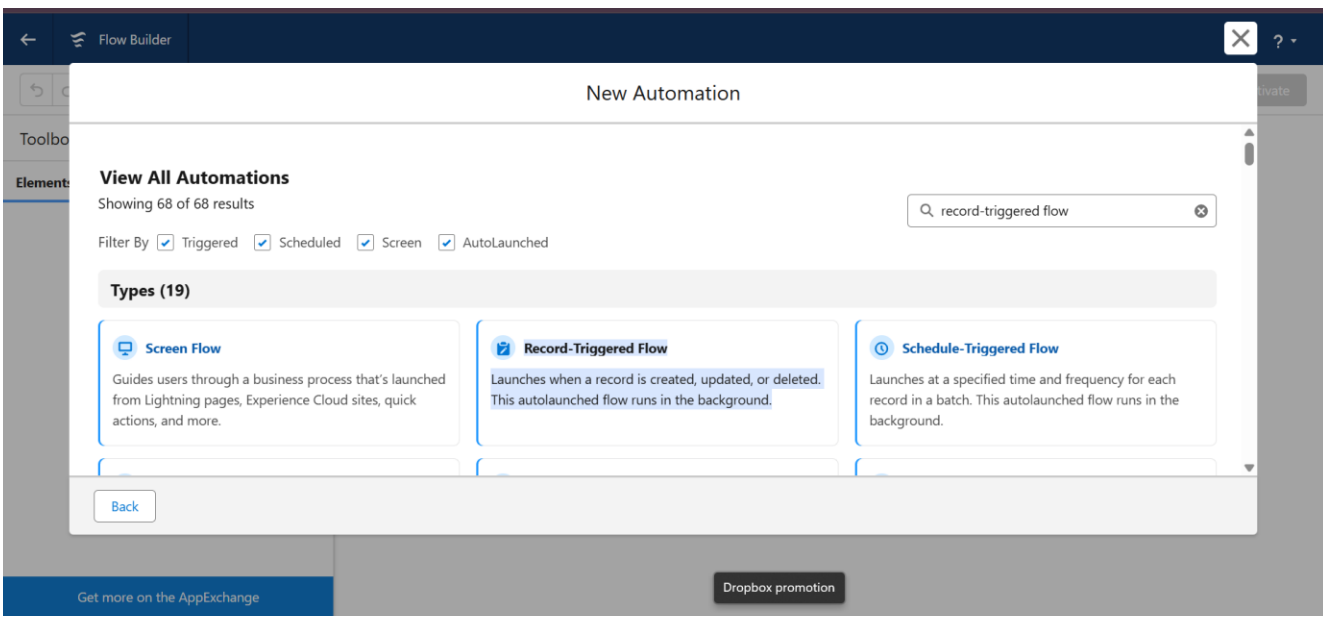
MILESTONE 13: Flows

ACTIVITY 1: Create a flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.



2. Select the Record-triggered flow and Click on Create.



3. Select the Object as “Billing details and feedback”in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.
7. Give the Label Name : Amount Update
8. Api name : is auto populated
9. Set a filter condition : All Conditions are met(AND)
10. Field : Payment_Status__c

11. Operator : Equals
12. Value : Completed
13. And Set Field Values for the Billing details and feedback Record
14. Field : Payment_Paid__c
15. Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}
16. Click On Done.
17. Before creating another Element. Create a New Resource form Toolbox form top left.
18. Click on the New Resource, And select Variable.
19. Select the resource type as text template.
20. Enter the API name as “ alert”.
21. Change the view as Rich Text ? View to Plain Text.
22. In body field paste the syntax that given below.

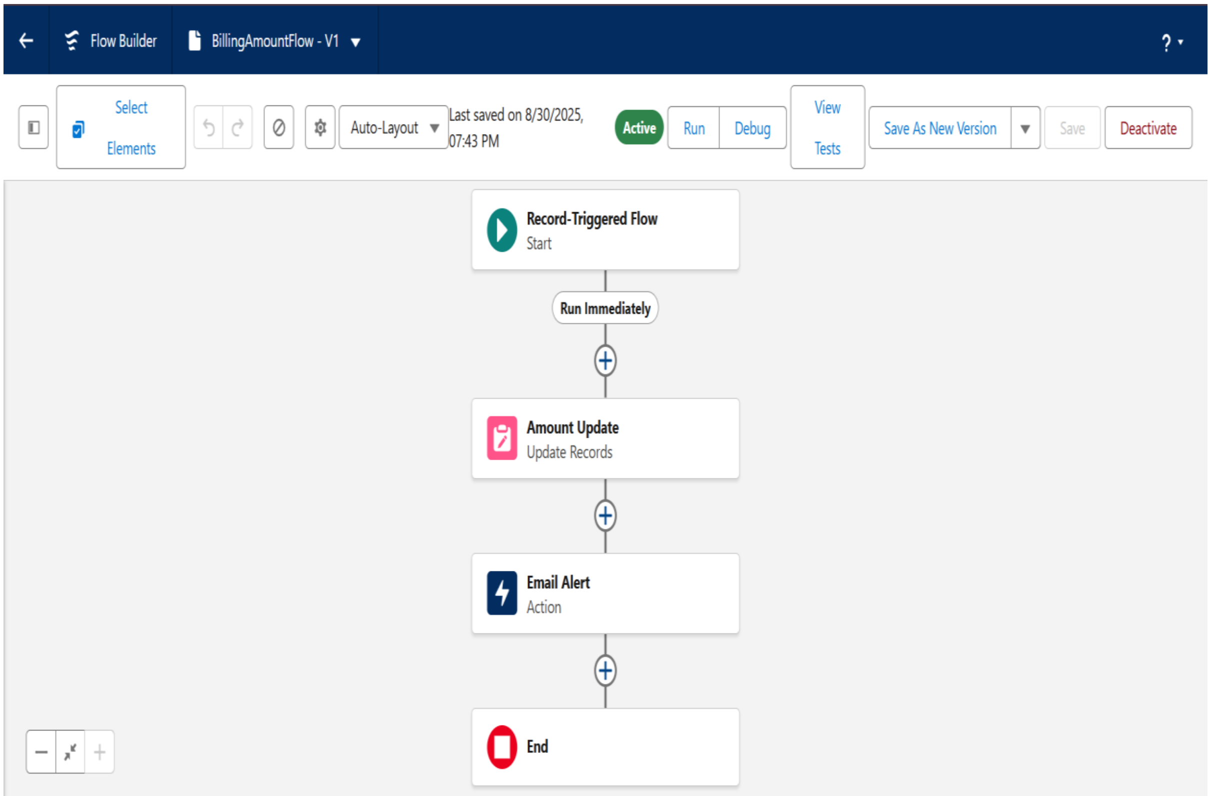
Dear {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!$Record.Payment_Paid__c}
Thank you for Coming .

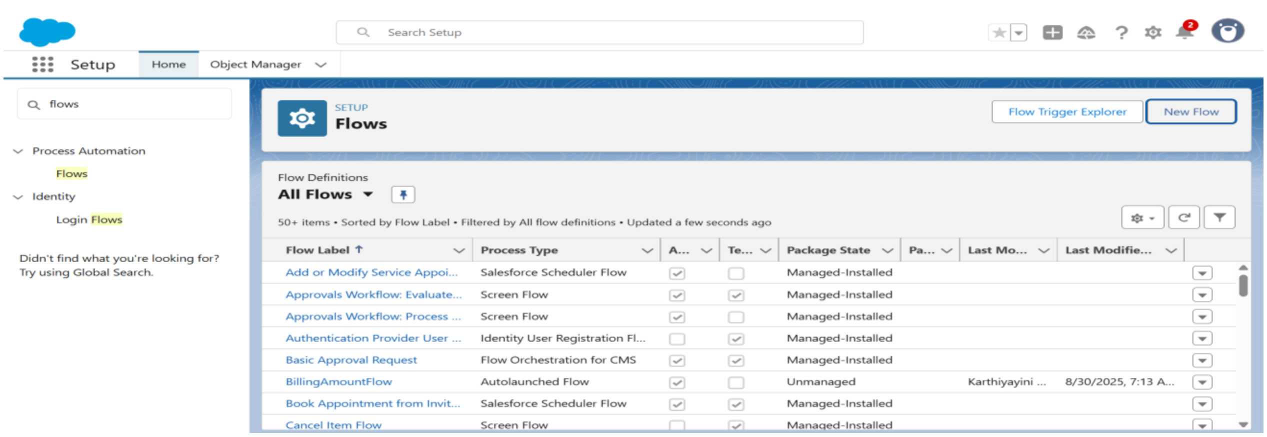
23. Click done.
24. Now Click on Add Element , select Action.
25. Their action bar will be opened in that search for “ send email ” and click on it.
26. Give the label name as “ Email Alert”
27. API name will be auto populated.
28. Enable the body in set input values for the selected action.
29. Select the text template that created , Body : {!alert}
30. Include recipient address list select the email form the record.
31. RecipientAddressList: {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
32. Include subject as “ Thank You for Your Payment - Garage Management”.
33. Click done.

34.Click on save. Give the Flow label , Flow Api name will be autopopulated.
35.And click save, and click on activate.

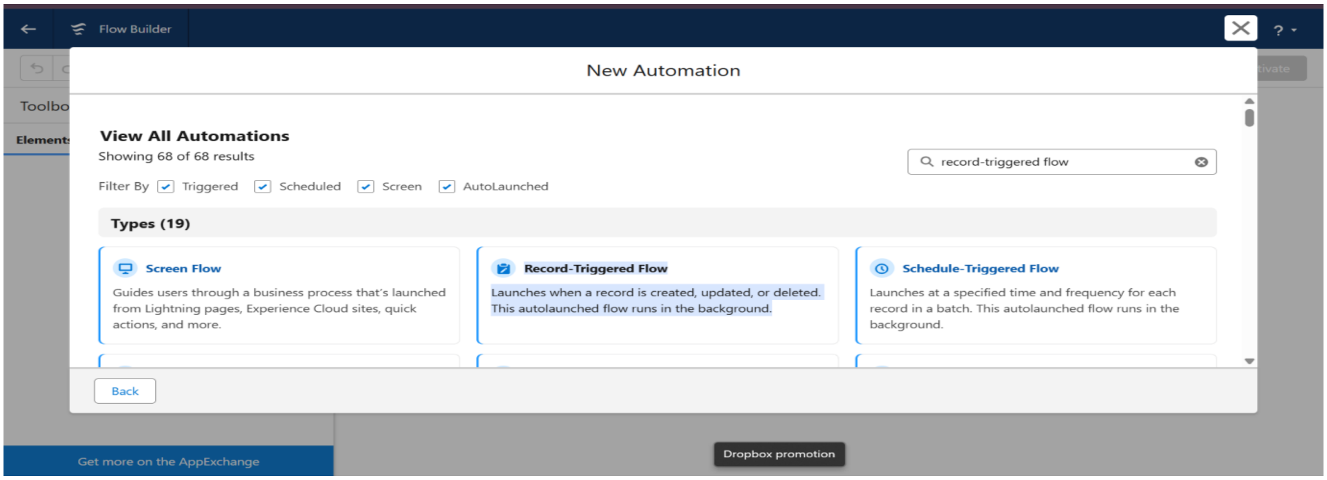


ACTIVITY 2: Create another flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.



2. Select the Record-triggered flow and Click on Create.



3. Select the Object as " **Service records**"in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimise the flow for: "Actions and Related Records" and Click on Done.
6. Under the Record-triggered Flow Click on "+" Symbol and In the Drop down List select the "Update records Element".

7. Set a filter condition : All Conditions are met(AND)

8. 8.Field : **Quality_Check_Status__c**

9. Operator : **Equals**

10. Value : **True**

11. And Set Field Values for the Billing details and feedback Record

12. Field : **Service_Status__c**

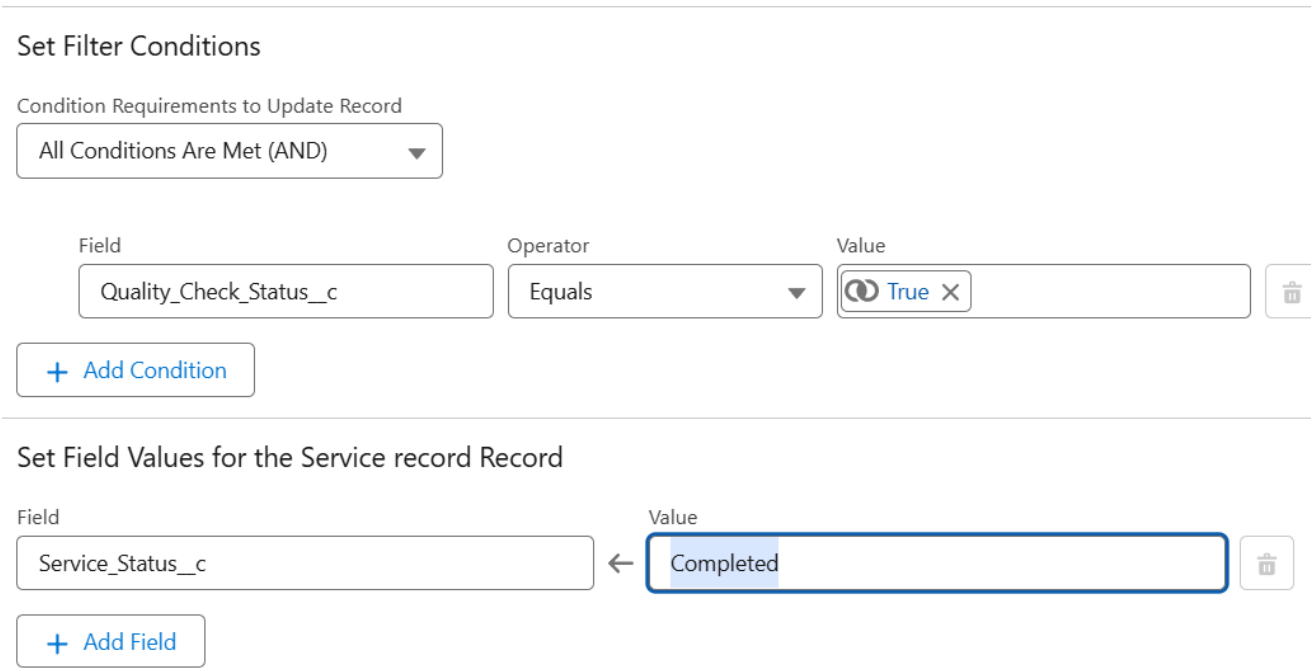
13. Value : **Completed**

14. Click On **Done**.

15. Click on **save**

16. Given the Flow label as **Update Service Status** , Flow Api name will be auto populated.

17. And click save, and click on **activate**.

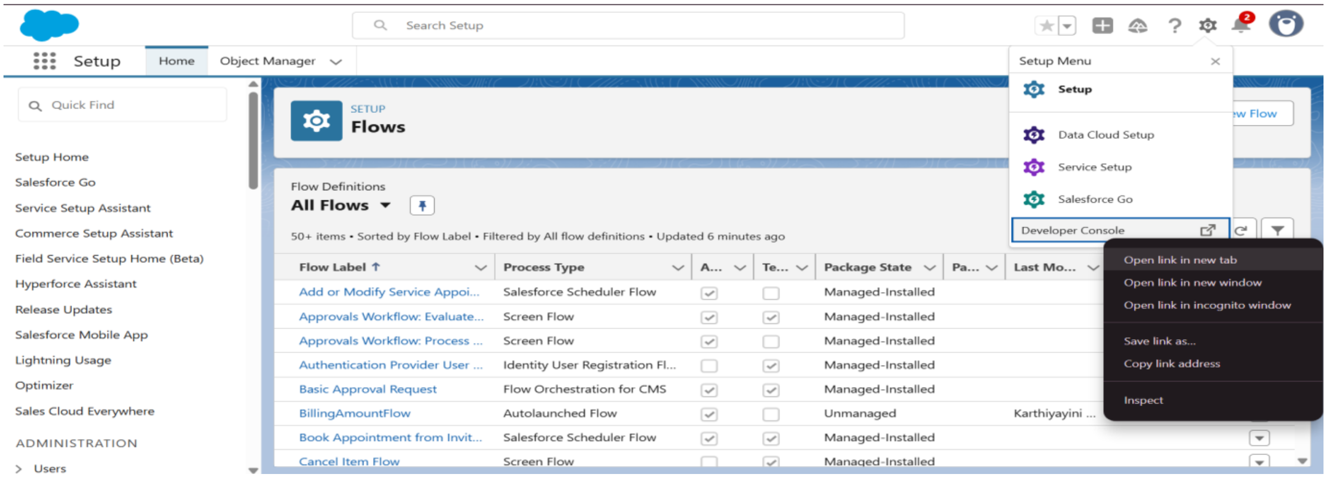


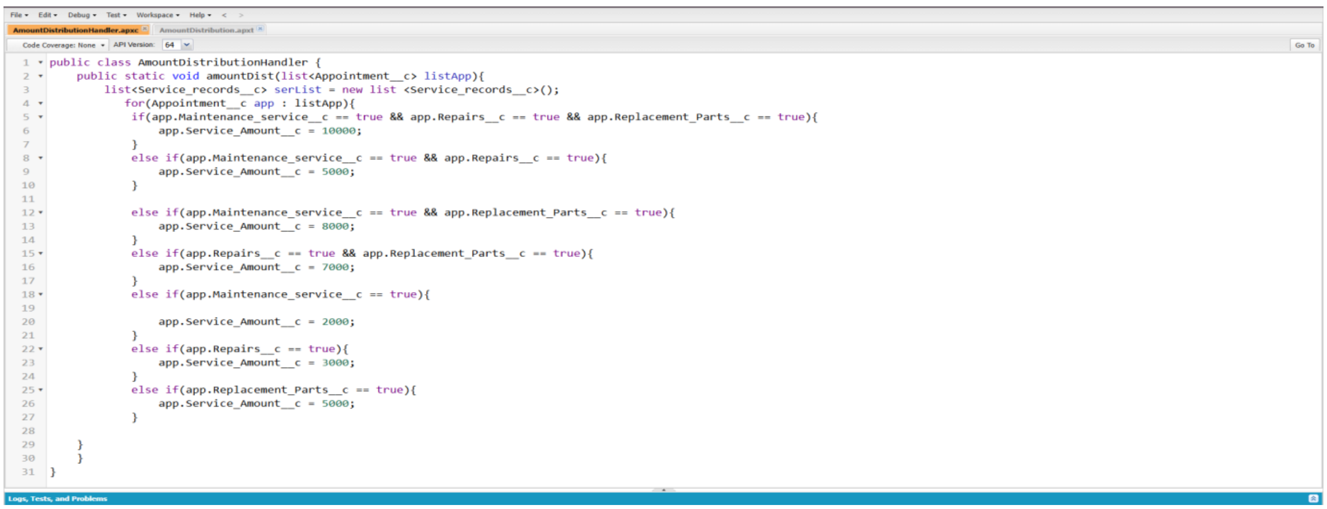
MILESTONE 14:Apex Trigger

ACTIVITY 1: Apex Handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as "AmountDistributionHandler ".





Code:

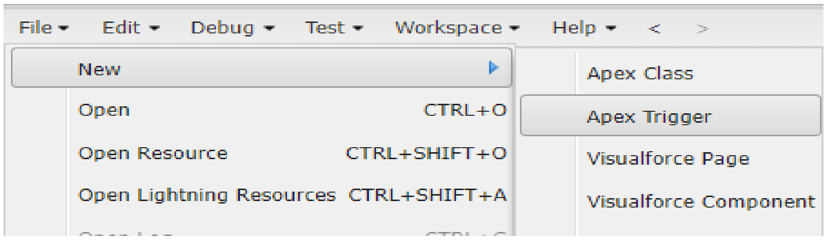
```
public class AmountDistributionHandler {

    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();

        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
```

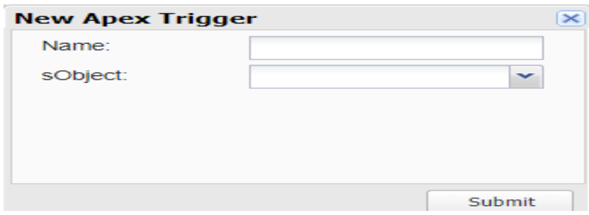
```
        }
        else if(app.Maintenance_service__c == true && app.Replacement_Parts__c ==
true){
          app.Service_Amount__c = 8000;
        }
        else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
          app.Service_Amount__c = 7000;
        }
        else if(app.Maintenance_service__c == true){
          app.Service_Amount__c = 2000;
        }
        else if(app.Repairs__c == true){
          app.Service_Amount__c = 3000;
        }
        else if(app.Replacement_Parts__c == true){
          app.Service_Amount__c = 5000;
        }

    }
    }
}
```

Trigger Handler :

How to create a new trigger :

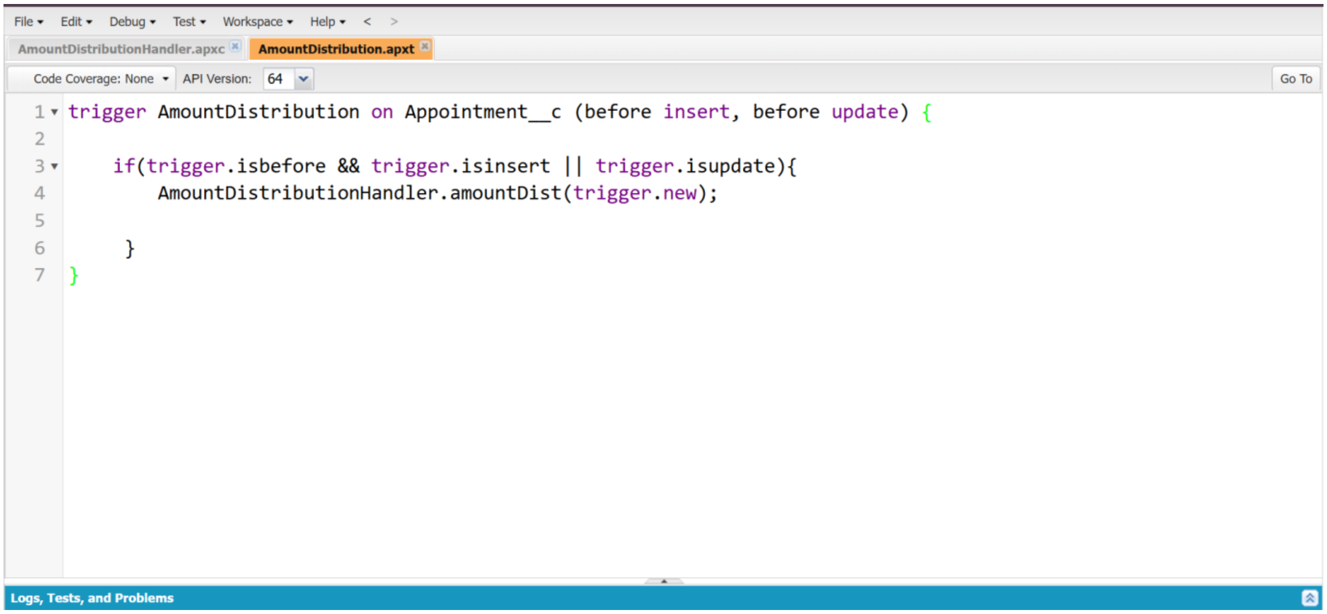
1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment__c





Syntax For creating trigger :
The syntax for creating trigger is :
Trigger [trigger name] on [object name](Before/After event)
{
}
In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1. Handler for the Appointment Object



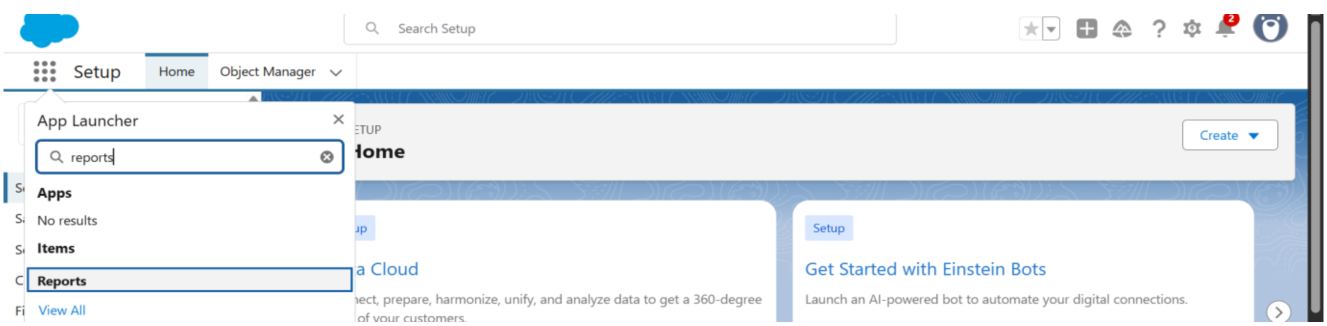
Code:

```
trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
      AmountDistributionHandler.amountDist(trigger.new);
    }
}
```

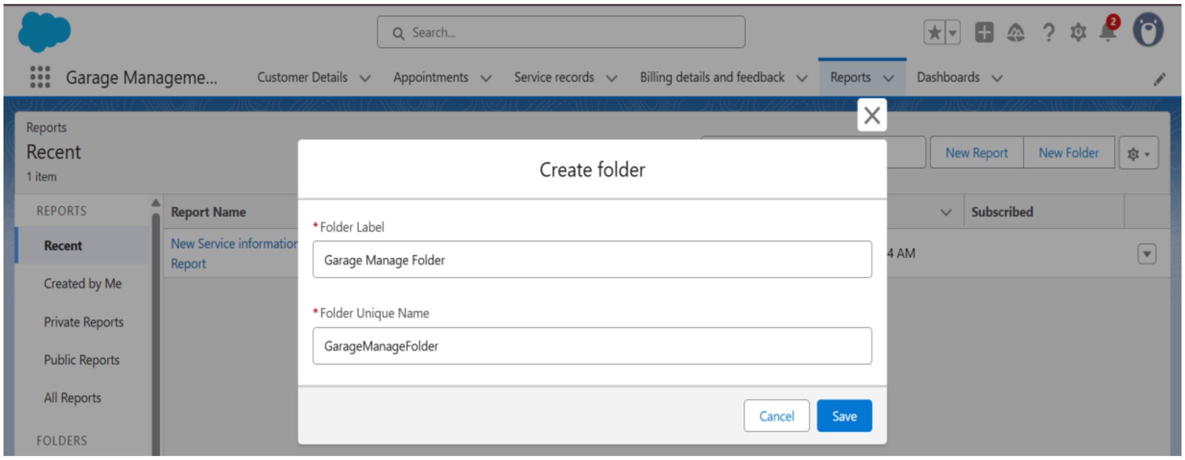
MILESTONE 15:Reports

ACTIVITY 1: Create a report folder

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.

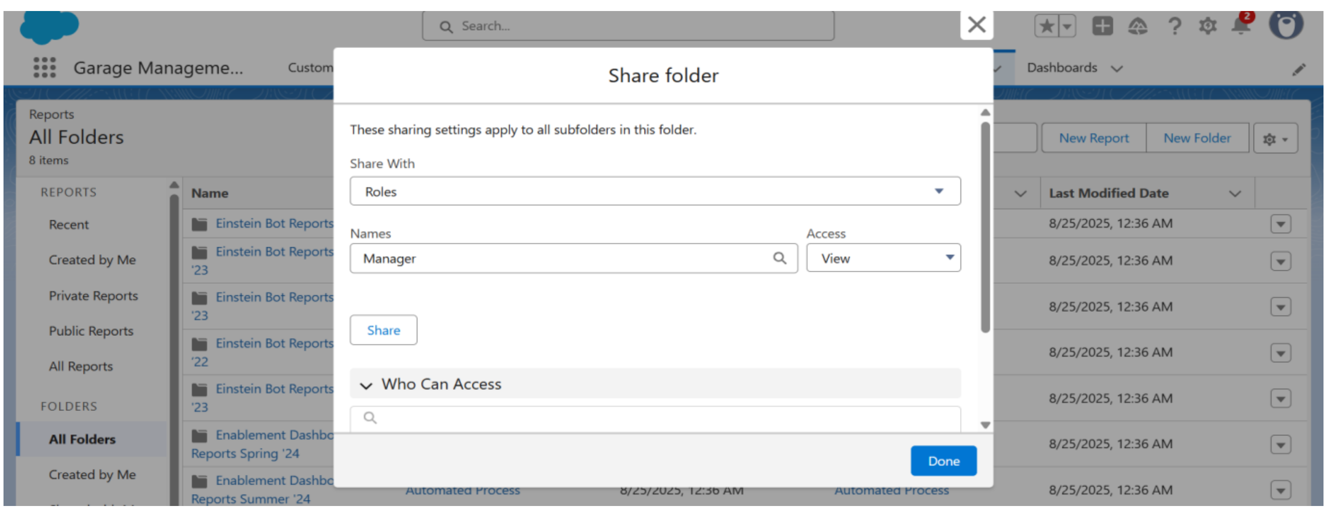


3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
4. Click save.



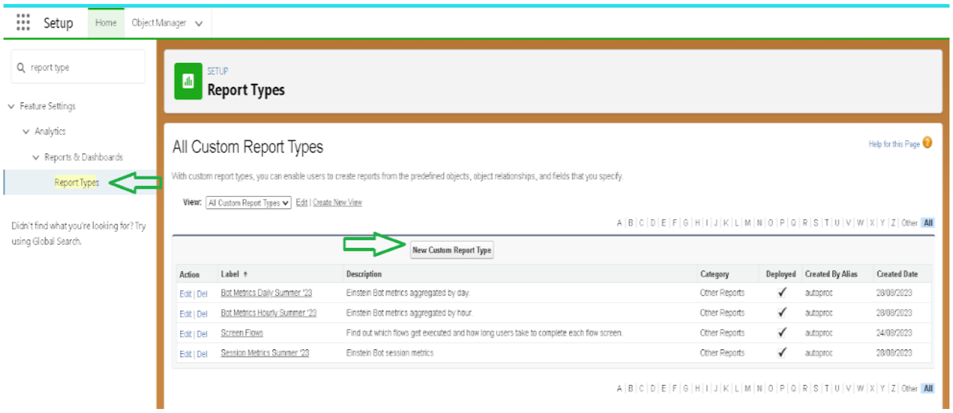
ACTIVITY 2: Sharing a report folder

1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.

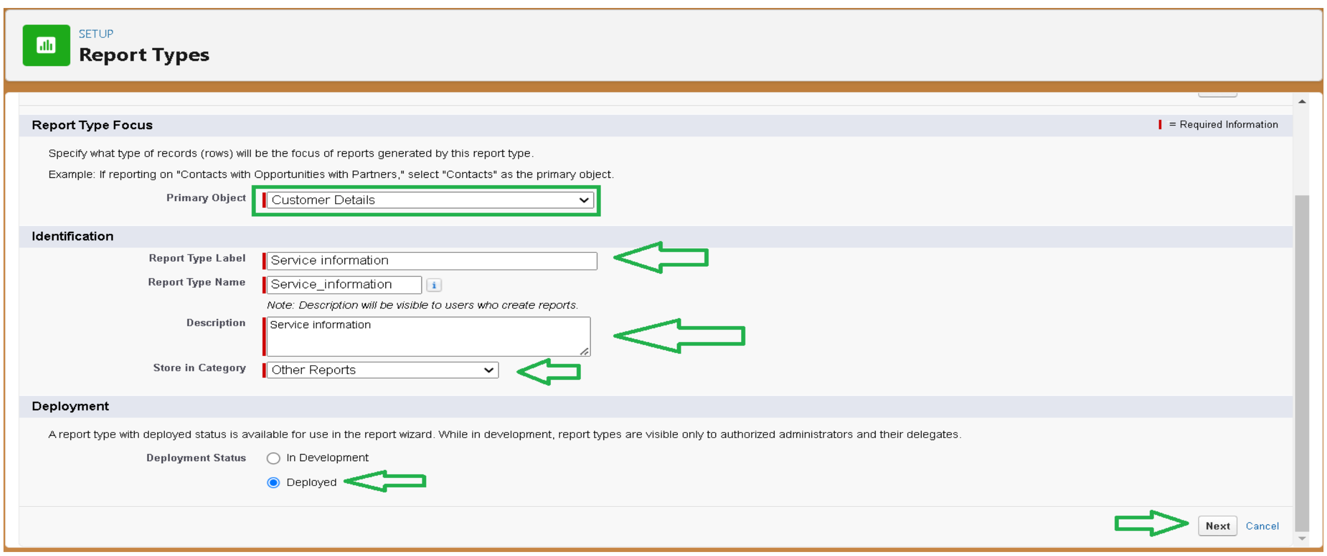


ACTIVITY 3: Create a report type

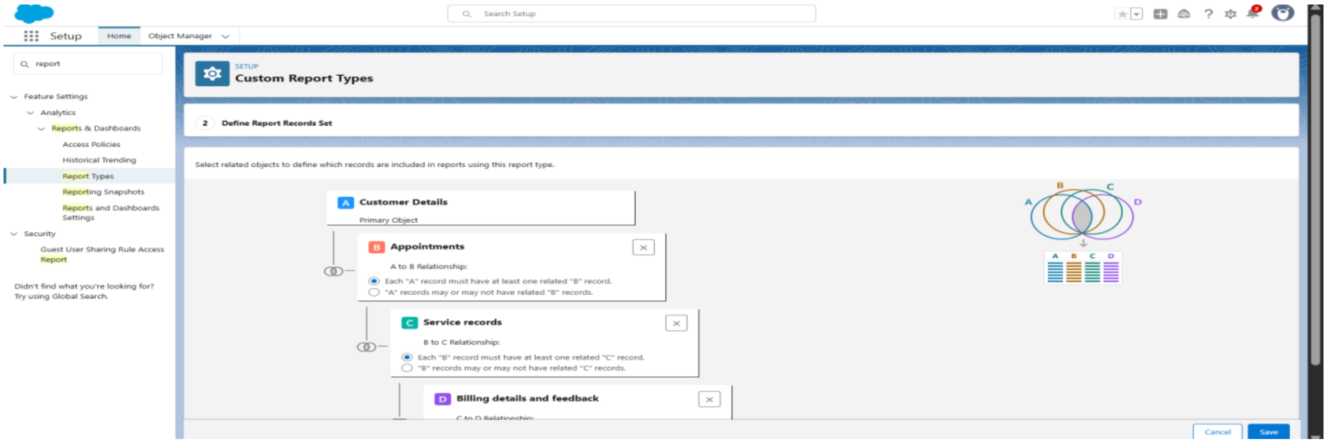
1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.



3. Select the Primary object as “ Customer details” .
4. Give the Report type Label as “ Service information ”
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as “ other Reports ”
8. Select the deployment status as “ Depolyed ”, click on Next.



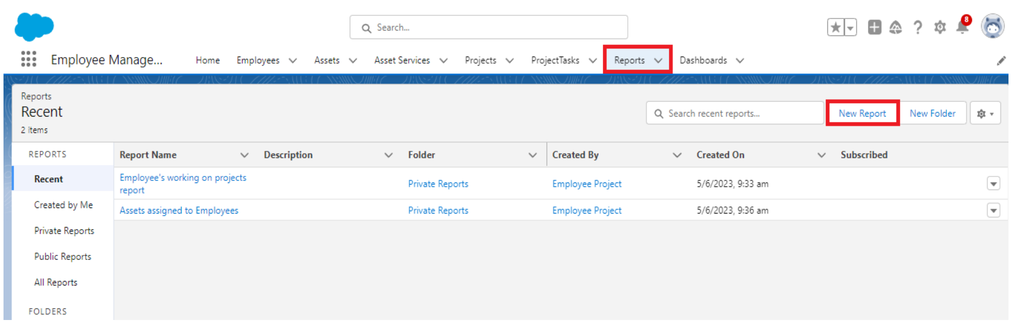
9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object as
11. Again Click to relate another object.
12. And select the related object as “ service records”.
13. Repeat the process and select the related object as “ Billing details and feedback”.
14. And click on save.



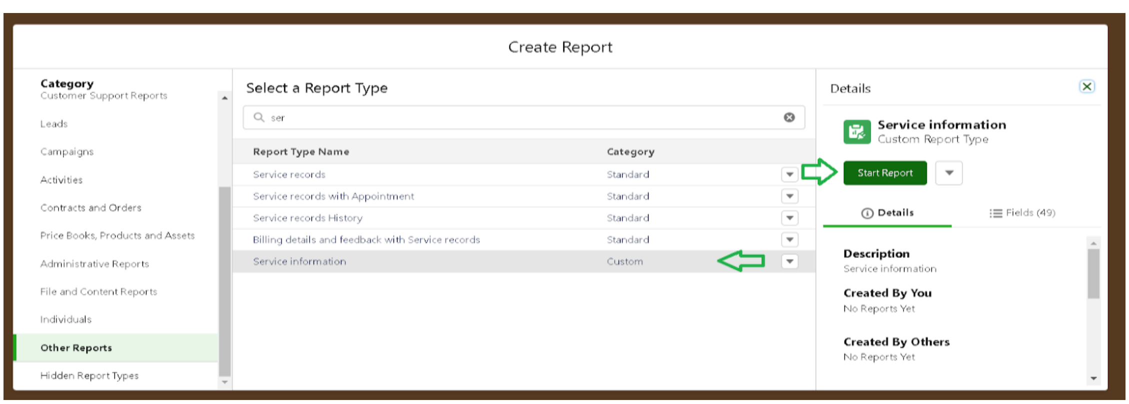
ACTIVITY 4: Create a report

Note : Before creating report, create latest "10" records in every object. Try to fill every field in each record for better experience.

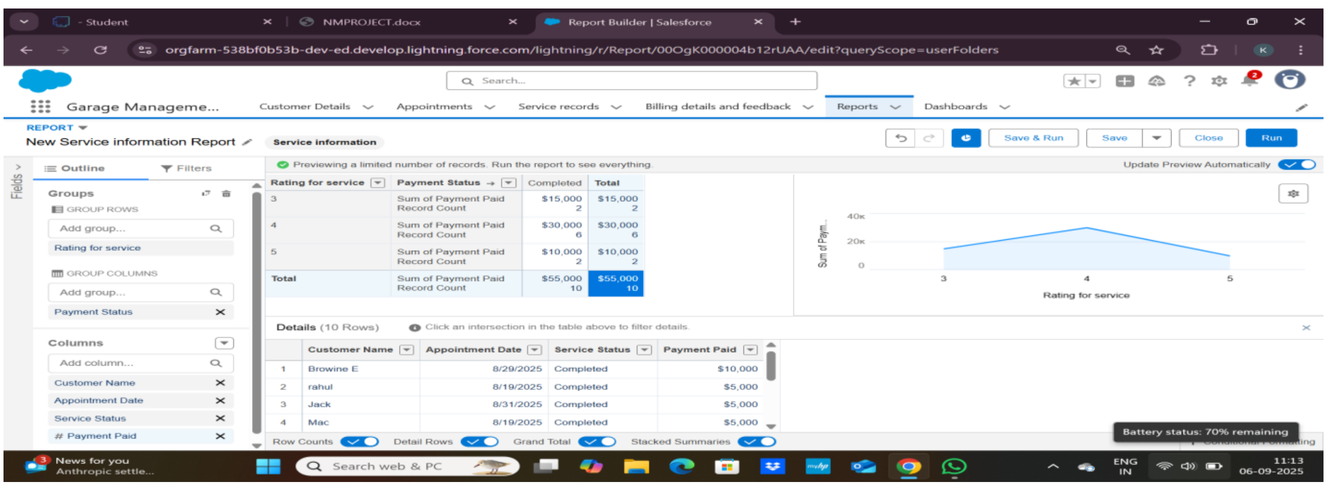
1. Go to the app >> click on the reports tab
2. Click New Report.

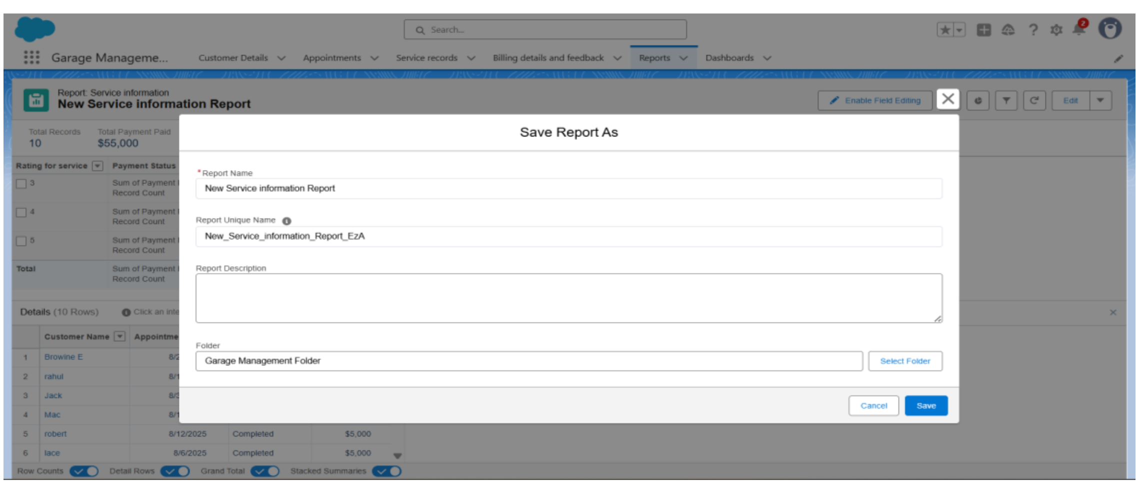


3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.



4. Their outline pane is opened alredy, select the fields that mentioned below in column section.
    a. Customer name
    b. Appointment Date
    c. Service Status
    d. Payment paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
    a. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
    a. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.

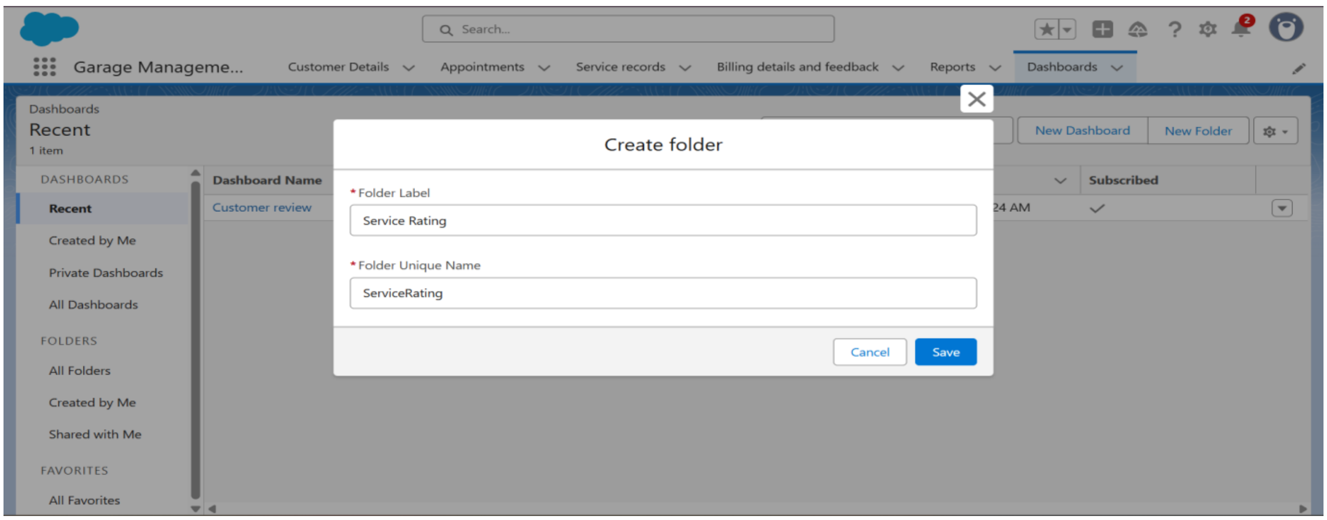




MILESTONE 16:Dashboards

ACTIVITY 1:Create a dashboard folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.

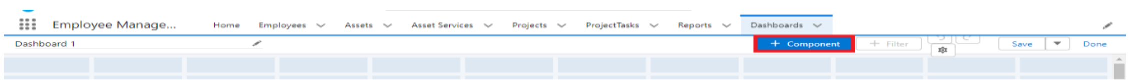


6. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

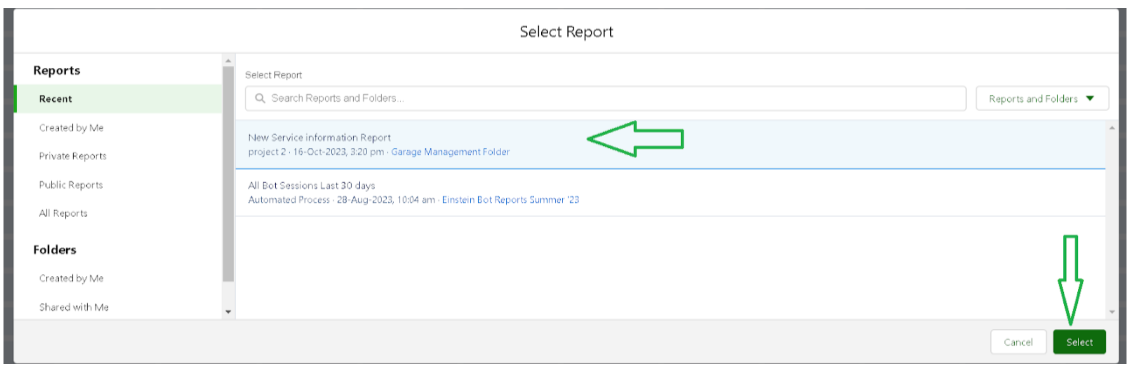
ACTIVITY 1:Create dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.

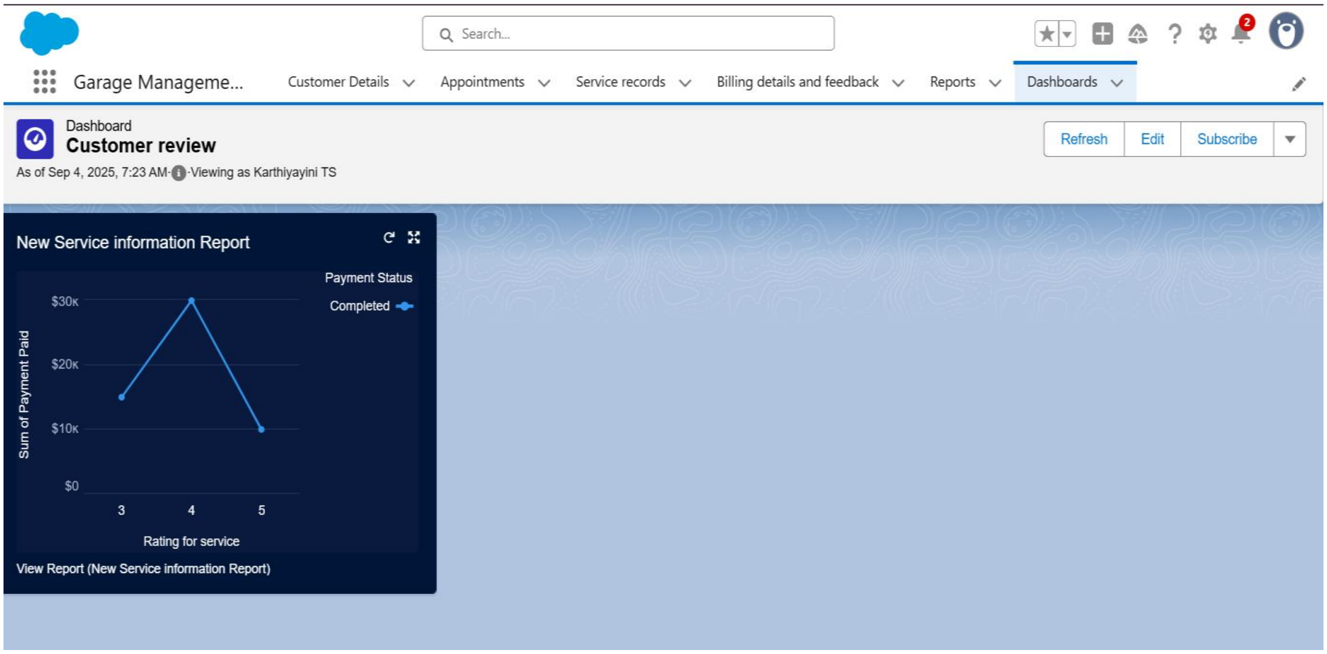
3. Select add component.



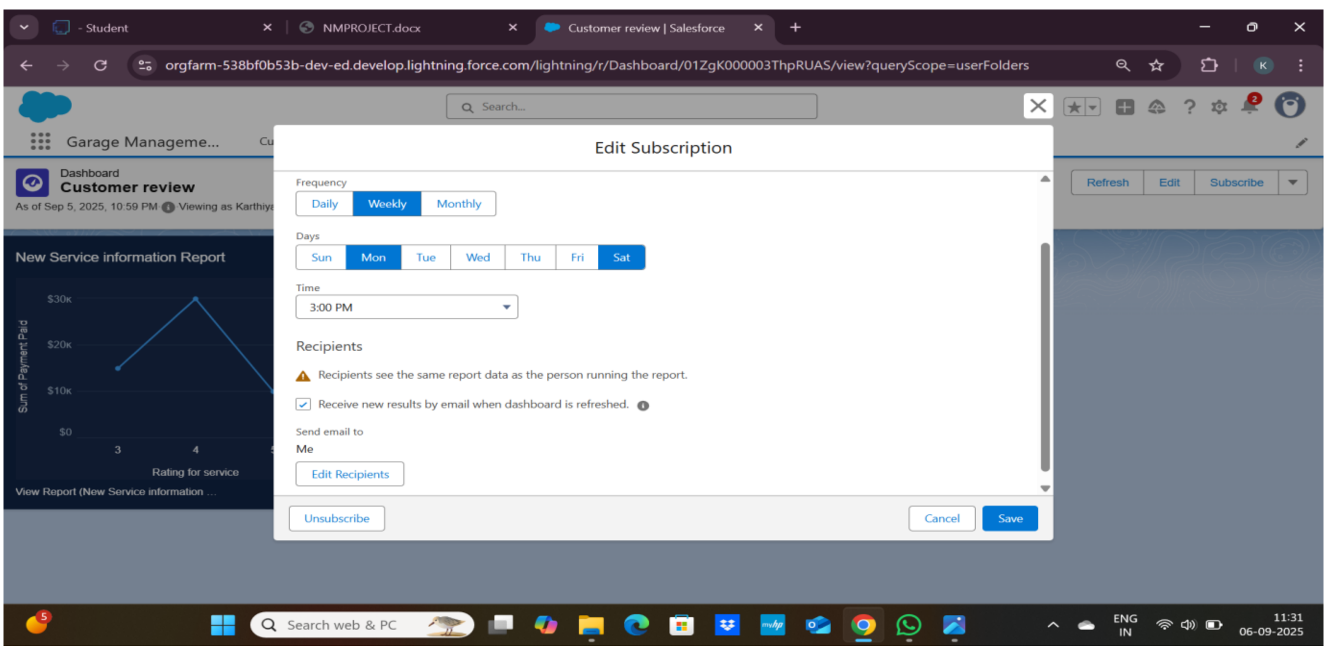
4. Select a Report and click on select.



5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.



8. After that Click on Subcribe on top right.
9. Set the Frequency as " weekly ".
10. Set a day as monday.
11. And Click on save.

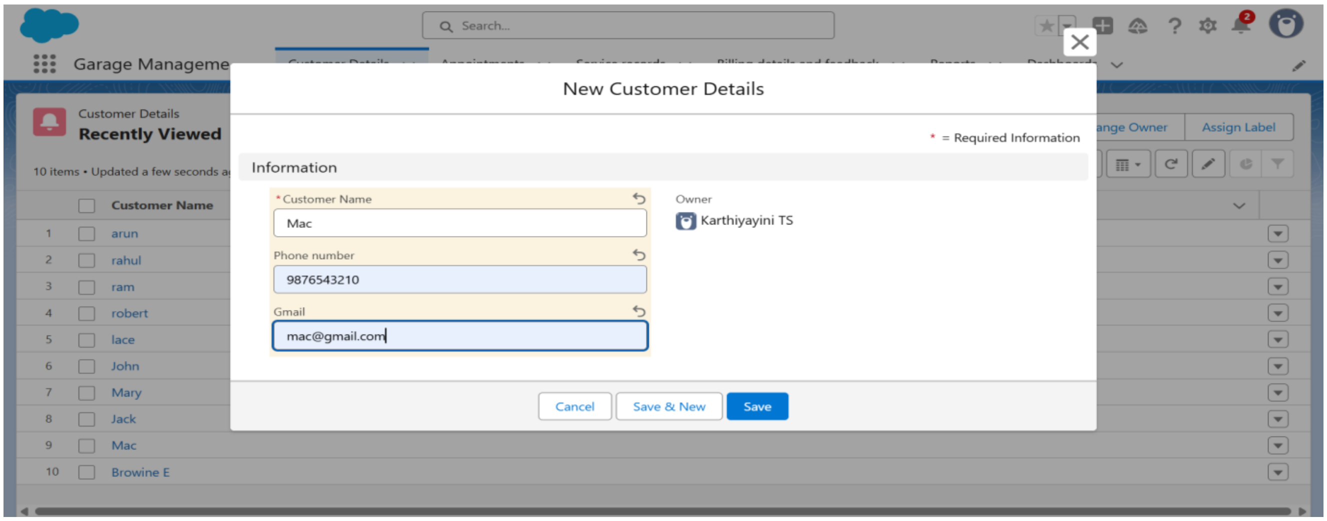


MILESTONE 17: User Adoption

ACTIVITY 1: Creating records

To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for " **Garage Management**" and click on it.
3. Click on the " **Consumer details** tab".
4. Click on new and fill the details as shown below figs, and click save.



Now, Create the Appointment Record

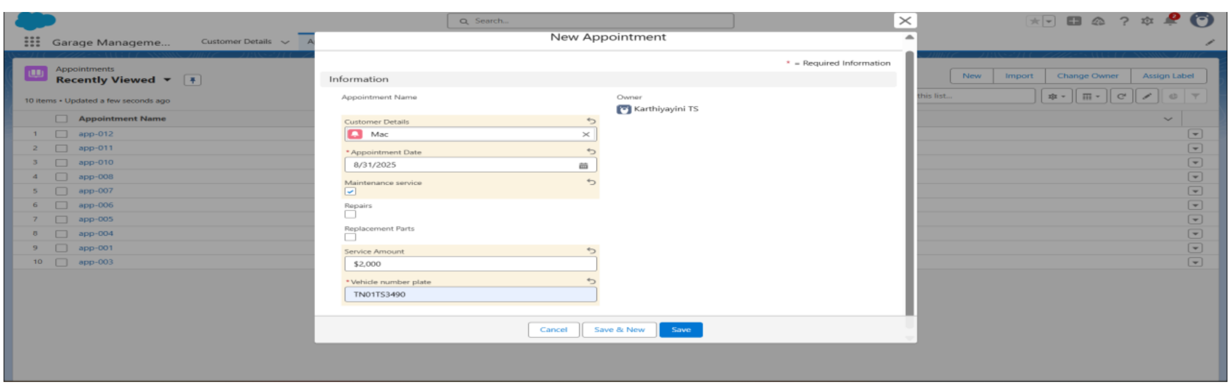
1.Click on the "**Appointment** tab".

2.Enter the customer details as created, while entering Appointment Date enter the date less than the created date.

3.Match the validation while entering the vehicle number plate.

4.Select the services you need.

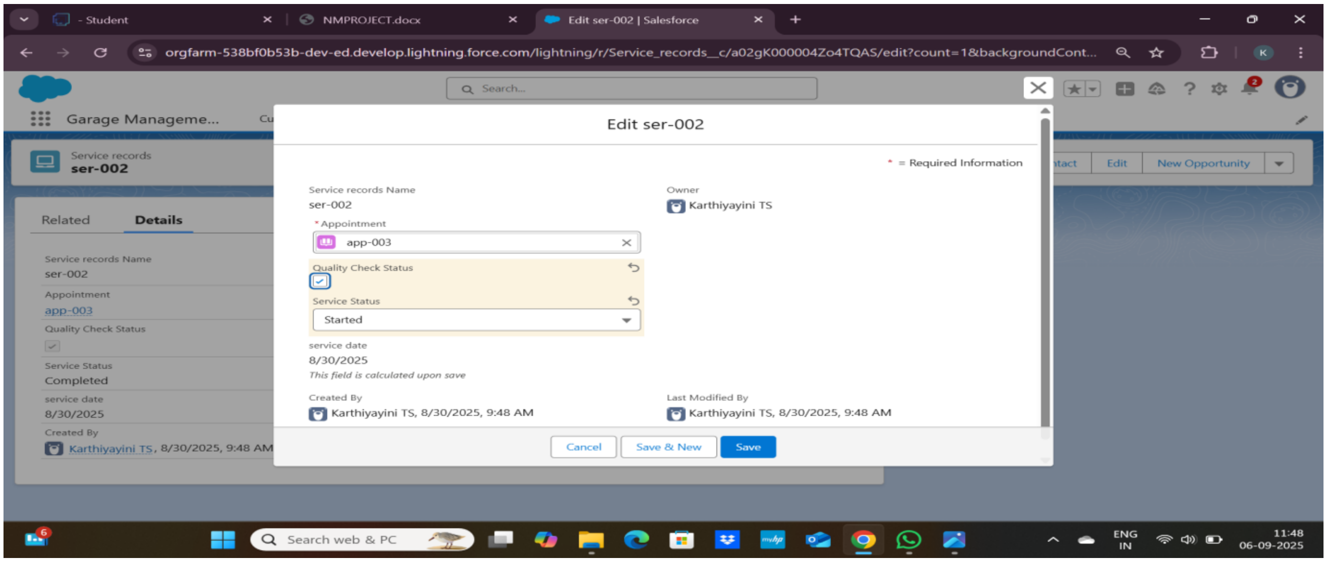
5.Click on save to see the Service Amount.



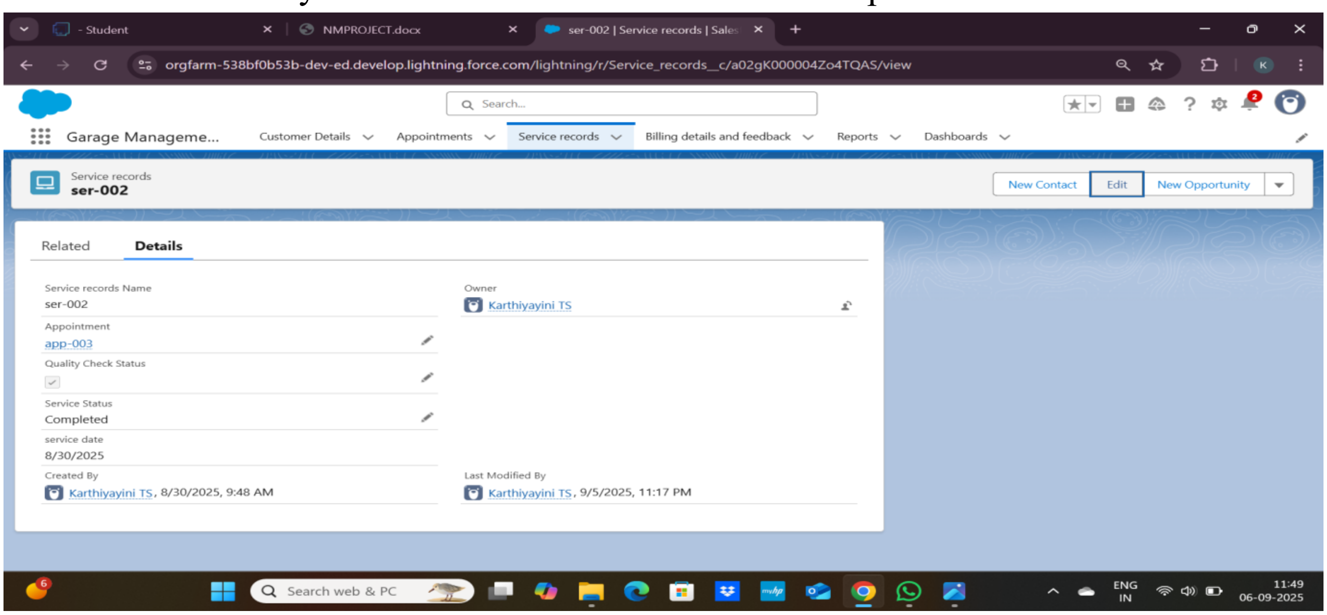
Now, Create a service Record

1. Click on the "**Service record** tab".
2. Enter the Appointment, and started is selected as default.
3. Click on save.

4. Open the record and click on Quality check status as true.
5. Click on save.



6.Now automatically Service status will be moved to completed.



NOTE : Thus all the milestone are created successfully using the above instuctions.

PHASE 3

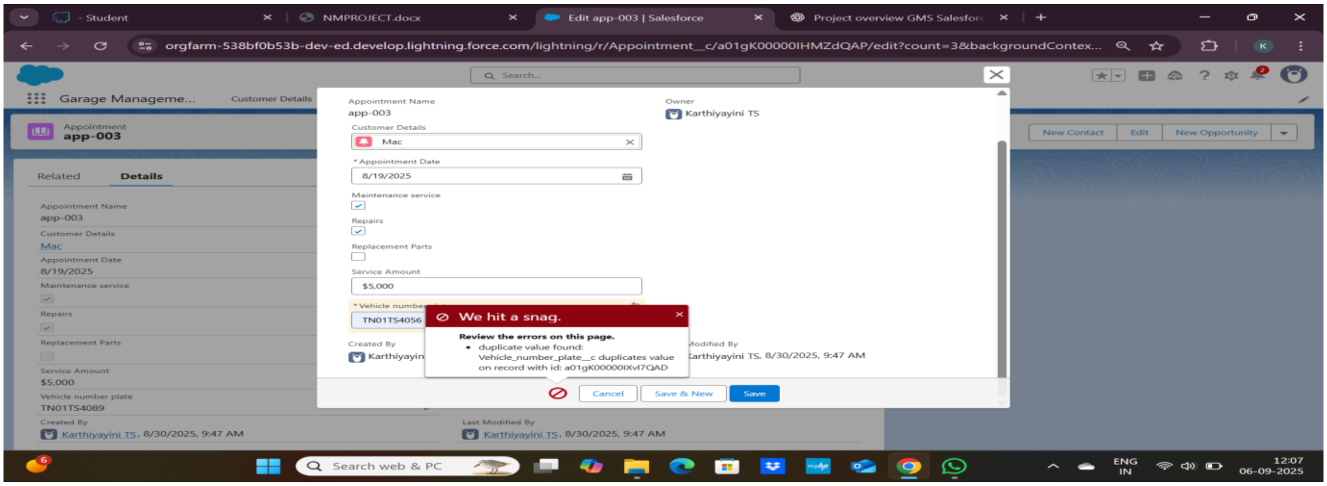
TESTING PHASE

FUNCTIONAL AND NON-FUNCTIONAL TESTING

Performance Testing

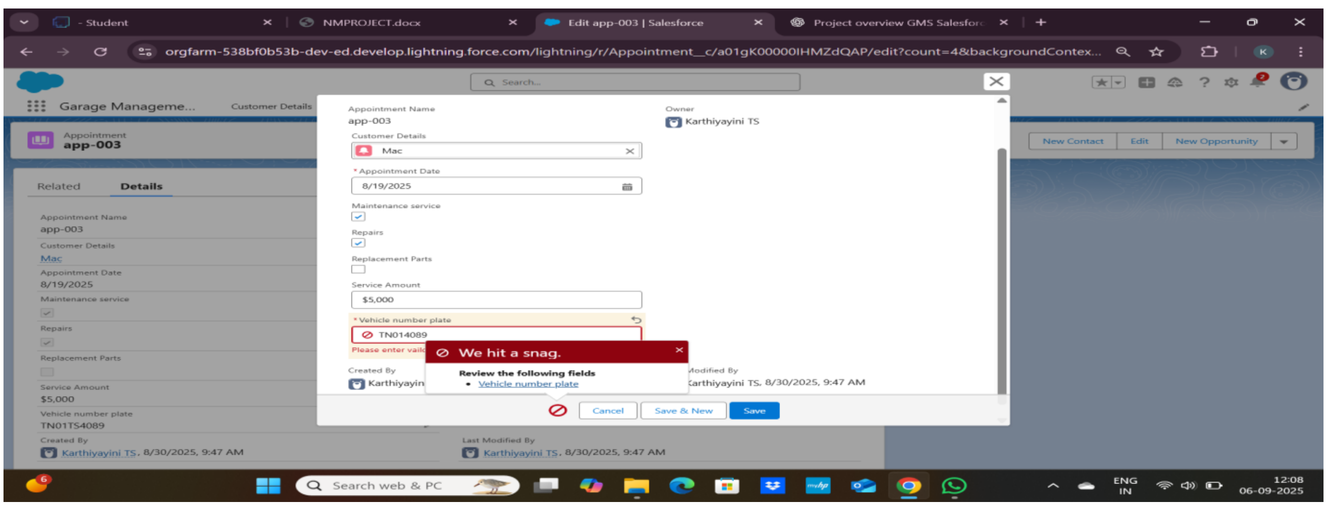
❖ **Duplicate Vehicle Record:**

Trigger validation by entering duplicate vehicle number records for the same customer.



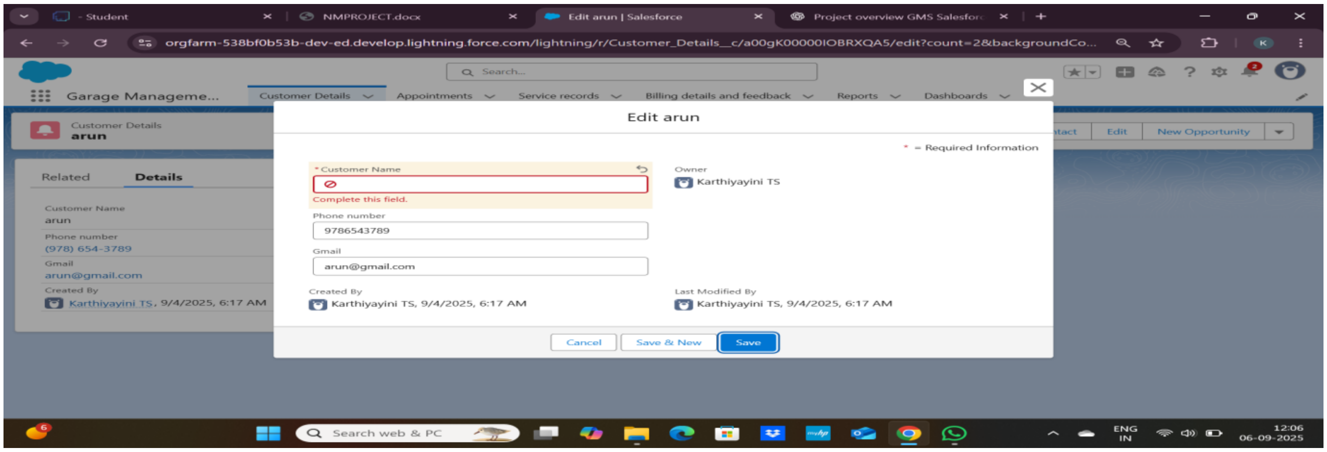
❖ **Invalid Vehicle Number Format:**

Trigger validation by entering a wrong or invalid number plate format.



❖ **Empty Customer Details:**

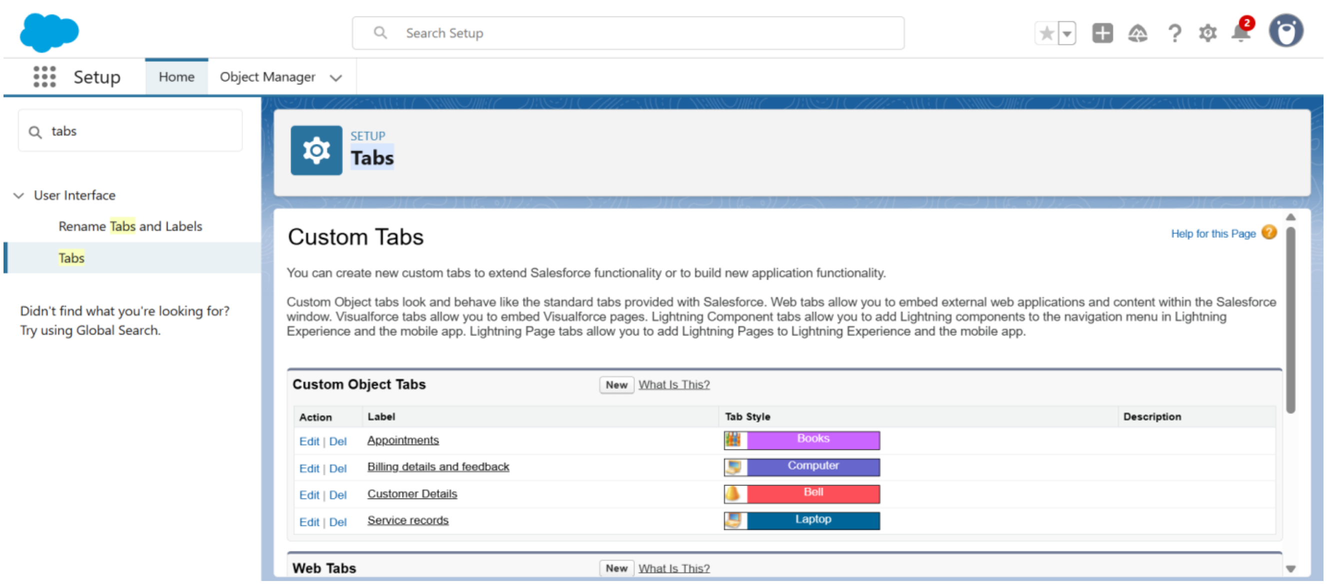
Trigger validation by leaving mandatory fields like customer name or contact number blank.



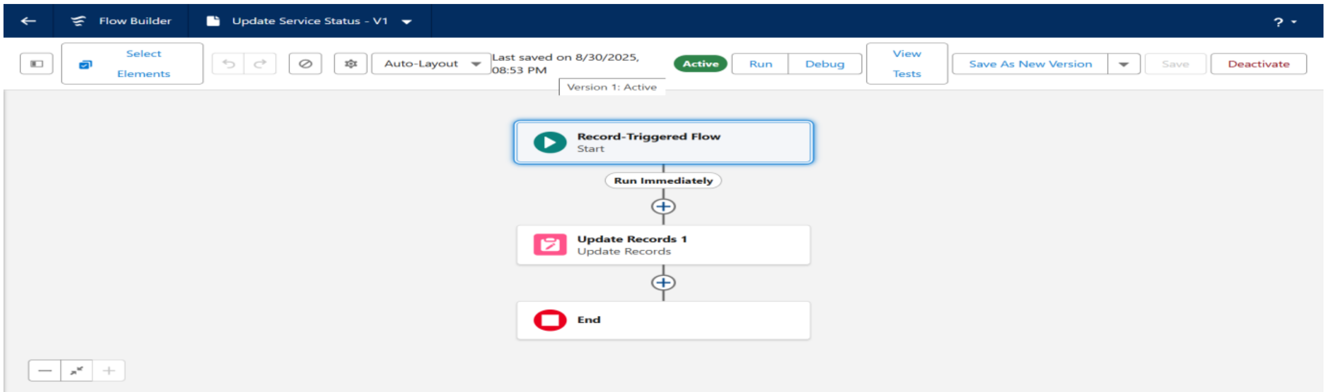
RESULT

OUTPUT SCREENSHOTS

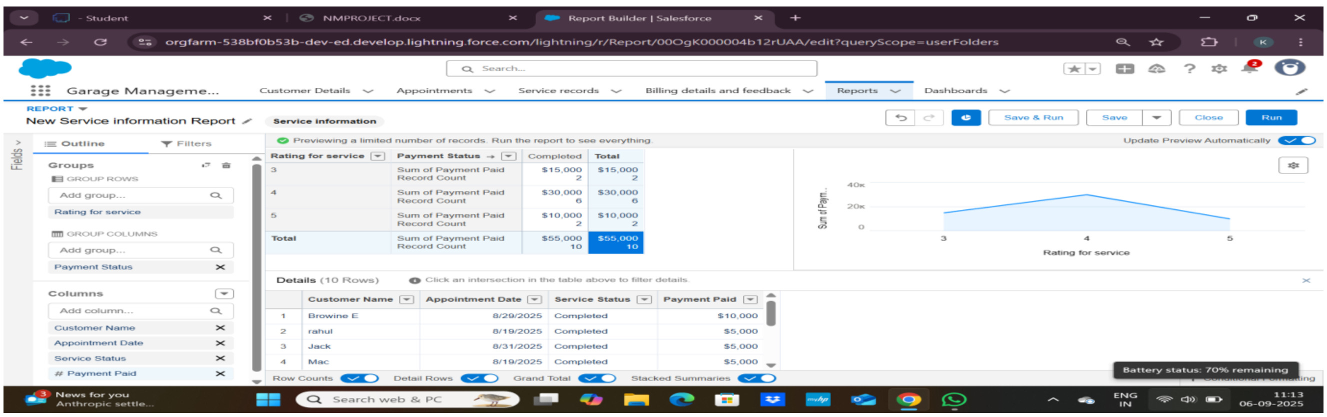
- Tabs for Appointments,Billing details and feedback , customer details and service records.



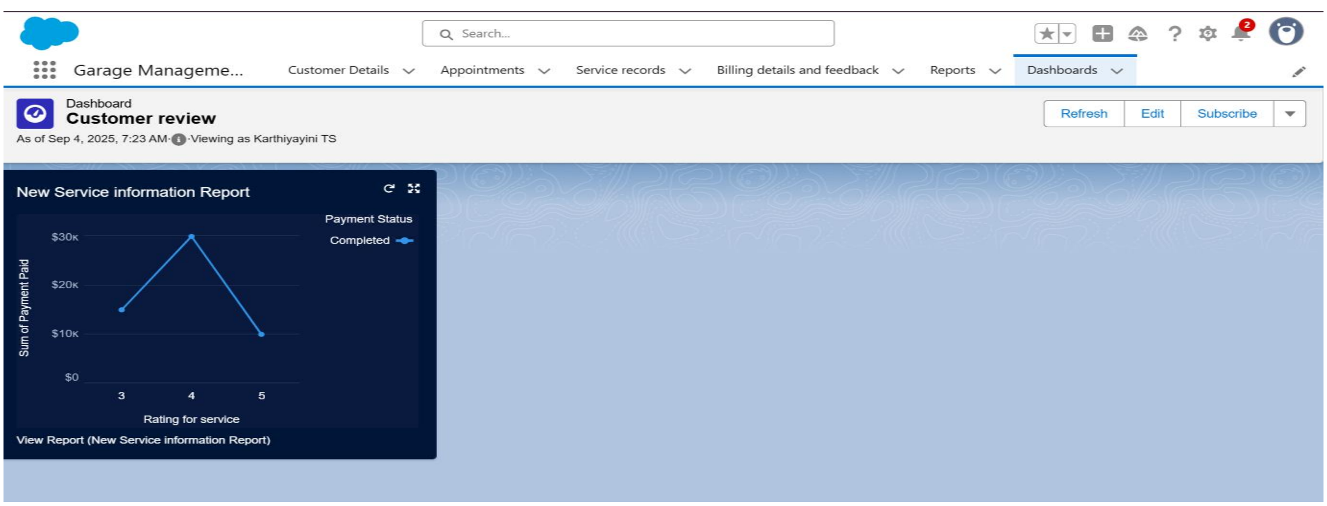
- Flows runs



- Reports



- Dashboards and the output line chart



CONCLUSION

The Garage Management System project demonstrates how Salesforce CRM can be applied to optimize garage operations. By automating tasks such as service scheduling, inventory tracking, and billing, the solution improves both efficiency and customer experience. This project highlights the importance of cloud-based solutions in transforming traditional business operations into scalable, data-driven systems.

- **Mobile Application Integration:** Extend the system with a mobile app for customers to book services, track status, and receive notifications in real time.
- **Payment Gateway Integration:** Enable secure online payments through UPI, debit/credit cards, and digital wallets.
- **AI-Powered Predictive Maintenance:** Use AI/ML in Salesforce to predict vehicle service needs and provide proactive reminders to customers.

APPENDIX

- **Source code:** Provided in Apex classes and triggers

Code for Apex classes

```
public class AmountDistributionHandler {
   public static void amountDist(list<Appointment__c> listApp){
       list<Service_records__c> serList = new list <Service_records__c>();
           for(Appointment__c app : listApp){
          if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
              app.Service_Amount__c = 10000;
          }
          else if(app.Maintenance_service__c == true && app.Repairs__c == true){
              app.Service_Amount__c = 5000;
          }
          else if(app.Maintenance_service__c == true && app.Replacement_Parts__c ==
true){
              app.Service_Amount__c = 8000;
```

```
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
             app.Service_Amount__c = 5000;
            }
        }
    }
}
```

Code for Apex triggers

```
trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }

}
```